11-4-2010

Feasibility study of a novel method for real-time aerodynamic coefficient estimation

Phillip Gurbacki

Feasibility Study of a Novel Method for Real-Time Aerodynamic Coefficient Estimation

by

Phillip M. Gurbacki

Thesis submitted to the Faculty of the Department of Mechanical Engineering in
the Kate Gleason College of Engineering at Rochester Institute of Technology
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Approved by:

Dr. Agamemnon Crassidis - *Thesis Advisor* _____
Department of Mechanical Engineering

Dr. Tuhin Das _____
Department of Mechanical Engineering

Dr. Jason Kolodziej _____
Department of Mechanical Engineering

Dr. Edward Hensel _____
Department Head of Mechanical Engineering

Rochester, New York
November 4, 2010

Keywords: Real-Time Estimation, Parameter Estimation, Sliding Observer,
Lyapunov Estimation, Nonlinear System Identification, Sliding Mode Control

Permission to Reproduce Thesis

Feasibility Study of a novel method for Real-Time Aerodynamic Coefficient Estimation

I, Phillip M. Gurbacki, hereby grant permission to the Wallace Memorial Library of Rochester Institute of Technology to reproduce my thesis in the whole or part. Any reproduction will not be for commercial use or profit.

Date: _____ Signature: _____

November 4, 2010

Acknowledgements

I would like to thank all of the faculty and staff members in the Mechanical Engineering department at the Rochester Institute of Technology for providing me with an exceptional educational experience. Specifically, I would like to thank Dr. Agamemnon Crassidis for not only for being my thesis advisor, but for his continued support throughout my career at RIT. Without his guidance and patience, this work and my success would not have been possible. I would also like to acknowledge both Impact Technologies, LLC and Moog, Inc whose guidance and funding have allowed me to complete the classes and research required for graduation. Finally, I would like to thank my family members and friends for their continued support and encouragement in all stages of my life.

Tommy: *"You know, a lot of people go to college for seven years."*
Richard: *"I know. They're called doctors."*

Tommy Boy. Dir. Peter Segal. Paramount Pictures, 1995.

Abstract

In this work, a feasibility study of a novel technique for the real-time identification of uncertain nonlinear aircraft aerodynamic coefficients has been conducted. The major objective of this paper is to investigate the feasability of a system for parameter identification in a real-time flight environment. This system should be able to calculate aerodynamic coefficients and derivative information using typical pilot inputs while ensuring robust, stable, and rapid convergence. The parameter estimator investigated is based upon the nonlinear sliding mode control schema; one of the main advantages of the sliding mode estimator is the ability to guarantee a stable and robust convergence. Stable convergence is ensured by choosing a sliding surface and function that satisfies the Lyapunov stability criteria. After a proper sliding surface has been chosen, the nonlinear equations of motion for an F-16 aircraft are substituted into the sliding surface yielding an estimator capable of identifying a single aircraft parameter. Multiple sliding surfaces are then developed for each of the different flight parameters that will be identified. Sliding surfaces and parameter estimators have been developed and simulated for the pitching moment, lift force, and drag force coefficients of the F-16 aircraft. Comparing the estimated coefficients with the reference coefficients shows rapid and stable convergence for a variety of pilot inputs. Starting with simple doublet and sin wave commands, and followed by more complicated continuous pilot inputs, estimated aerodynamic coefficients have been shown to match the actual coefficients with a high degree of accuracy. This estimator is also shown to be superior to model reference or adaptive estimators, it is able to handle positive and negative estimated parameters and control inputs along with guaranteeing Lyapunov stability during convergence. Accurately estimating these aerodynamic parameters in real-time during a flight is essential for advanced control systems that require the estimated parameters as inputs.

Contents

List of Figures

List of Tables

Nomenclature

α, β	true angle of attack and side-slip angle
$\bar{c}$, b, c	aircraft mean aerodynamic chord, wing span, and wing area
$\bar{q}$	dynamic pressure
$\bar{x}_{cg}$, $\bar{y}_{cg}$, $\bar{z}_{cg}$	aircraft center-of-gravity location relative to the aircraft reference point
δ_e, δ_a, δ_r	aircraft elevator, aileron, and rudder deflection
η	strictly positive gain used to prevent divide by zero condition in estimator
λ	strictly positive gain used in sliding surface of estimator
ϕ, θ, ψ	aircraft roll, pitch, and heading angles
C_D, C_L, C_m	aerodynamic drag force, lift force, and pitching moment coefficients
C_l, C_n, C_Y	aerodynamic rolling moment, yawing moment, and side force coefficients
g	acceleration due to gravity
h	altitude
I_{xx},I_{yy}, I_{zz}	moment of inertia about x, y, and z body fixed axes reference frames, respectively
I_{xy},I_{xz}, I_{yz}	products of inertia
m	mach number
n_x, n_y, n_z	components of acceleration along x, y, and z body fixed axes at c.g., respectively
p, q, r	components of angular velocity along x, y, and z body fixed axes, respectively
V	aircraft's true velocity

Chapter 1

Introduction

1.1 Intelligent Flight Control System

The Intelligent Flight Control System (IFCS) flight research project has been established at the NASA Dryden Flight Research Center to create a revolutionary technological breakthrough in aircraft flight controls that can efficiently optimize aircraft performance in both normal and faulty flight conditions. Modifying in real time the flight control laws of an aircraft that has been damaged in flight (battle damage, weather, or system failure) will enable a pilot to maintain control and safely land an aircraft that has suffered major damage. Along with maintaining control during a major failure the IFCS can immediately adjust to maintain the best possible flight performance during an unexpected failure. When an aircraft has a major control surface or airframe damage the flight control system's design integrity is compromised and can cause the aircraft to become uncontrollable or unstable. The primary goal of the IFCS project is the development of adaptive and fault-tolerant flight control systems leading to unprecedented levels of safety and survivability for both civil and military aircraft. Real-time parameter identification (PID) of the aerodynamic coefficients, stability, and control derivatives of the damaged aircraft is necessary for the correct updates to be made to the control laws to ensure proper aircraft operation under faulty conditions. One of the major drawbacks of the current IFCS system PID is the reliance on programmed excitation command signals being sent to each individual surface to determine control surface effectiveness. When implemented on a high performance fighter jet, designed to have a human pilot

on board, the excitation to the control system would have to override the pilot inputs for approximately ten seconds [1]. Once the aerodynamic coefficients are identified they are passed along to the adaptive online-learning neural network known as the Dynamic Cell Structure (DCS) [2]. The DCS continuously chooses a set of stability and control derivatives for the current flight condition and damage level, once this information is available it is passed to the reconfigurable flight control system. Figure 1.1 shows the real time PID role in the complete IFCS system, this figure shows the interaction of the identified aerodynamic coefficients with the entire IFCS [3].

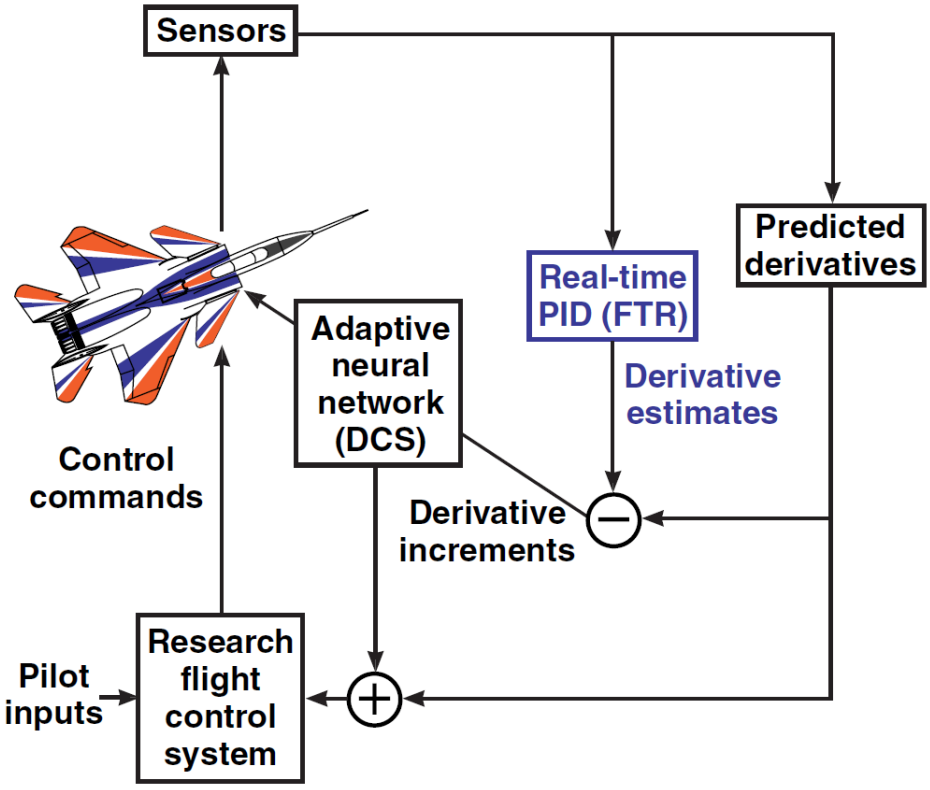

Figure 1.1: Schematic of the Real-Time PID Role for IFCS Research [1]

The goal of this work is to develop and implement a novel real-time PID algorithm that can be used by the IFCS or similar fault tolerant adaptive control systems. The algorithm will be developed for multiple aerodynamic coefficients on the Lockheed Martin F-16 aircraft and will ensure robust, stable, and rapid convergence. The control surface excitation used for parameter estimation will be generated from typical pilot commands, and will not need to have programmed excitation signals sent to the control surfaces.

Successful identification of aerodynamic parameters and integration with an adaptive control system will ensure proper control of the aircraft under adverse flight conditions.

1.2 Parameter Estimation

The sliding mode parameter estimator that is proposed is based on common sliding mode control theory. A standard sliding mode controller generally solves for the u (input) in the systems equation and determines the proper input which will linearize the system and provide for a more controllable plant. The use of sliding mode control as an estimator in this paper is being proposed as a means to identify parameters inside of a given system while guaranteeing Lyapunov stability and convergence. There has been a large amount of research in the field of parameter estimation, specifically with the use of sliding estimation schemes. The use of a sliding surface and lows pass filtering for on-line disturbance estimation has been successfully implemented in previous research and estimation results have been shown to be quite accurate and desirable [4]. The current use of sliding mode controllers and observers, where they are employed, and the implementation techniques used will be discussed in this section.

Sliding mode parameter estimation is distinctly different from sliding mode observers, both use the same underlying sliding mode control philosophy, however, sliding mode observers are generally used to observe system states, where parameter estimation is concerned with identifying a systems parameters. Sliding mode observers have been developed for fault reconstruction [5], rotor position and velocity estimation [6], and fault actuator estimation [7]. In all three of these cases sliding mode theory has been used to observe states of a given system, for the fault identification papers, the states observed were used in the fault detection and isolation in the system. A large focus of the previous work is on the situation when the system states are not all available and creating new developments in the use of sliding mode observer theory for decoupling the effects of fault signals from the response of the system estimated outputs. The example of a sliding observer shown in [6] observes the rotor position and velocity of a switched reluctance motor, in this case the systems states are identified through the use of a linear state-space model of the motor. The convergence of the observer is considered when there is an initial error in the position estimate and when there is an error in the flux

measurement. It is found that the sliding-mode observer is well suited to fixed-point arithmetic and that it has robust convergence properties under the conditions that arise in practical applications.

A variety of motor and rotating machinery applications have successfully incorporated a sliding mode controller and observer into their operation, the switching nature of the sliding observer proves to be useful in a continuously rotating environment [8][9][10]. These works show the derivations of sliding mode observers for motor and rotor flux applications, they also show robustness against modeling uncertainty and noise. The work that has been reviewed is significantly different than the parameter estimation technique as proposed in this work. There are many sliding observers that identify system states; however none of them follow the same derivation technique as proposed in Section 3.2.2.

1.3 Fault Tolerant and Adaptive Control

Intelligent flight control systems can do much to improve the operating characteristics of aircraft, and they provide an interesting, practical, and well documented framework within which intelligent control concepts can be developed. Adaptive control systems involve the modifying of control laws used by a controller to address changes in the system parameters, usually these changes can be slowly time-varying. For example, as an aircraft flies, its mass will slowly decrease as a results of fuel consumption, a control law that adapts to this change in fuel level is essential to achieve maximum performance from the aircraft [11]. Other types of parameter changes could be due to failures in the system, both slowly time-varying and instantaneous changes. Each of these cases require a measure of the changing variable, in the case of the fuel consumption a simple fuel level sensor would suffice, however in more complex systems a simple sensor will not provide the desired information. Where a sensor cannot be used, estimating parameters is a potential option for a measure of system change. The system capability to determine a change in parameter from nominal and then change the control system is the foundation of adaptive control. When the parameter is changed due to a system failure, the reconfiguration of the system based on the detection of the failure is considered a fault tolerant control system [12].

1.4 Overview and Motivation for Present Work

As aircraft characteristics have changed, and more importantly as the technology has allowed, an increased share of the aircraft's intelligent operation has relied on proper functioning of electromechanical sensors computers, and actuators [13]. It has become possible to apply machine intelligence to flight control. The transition toward automating control intelligence has been evolutionary, with adaptive control systems in use for a long time and advancing toward state of the art complete autonomous UAV's [14]. The parameter estimation for aircraft aerodynamic coefficients presented in this work will provide a great foundation for the development of an aircrafts adaptive and fault tolerant control system. The understanding of how the aircraft is currently behaving compared to the expected behavior provides a tremendous amount of information to an intelligent system and will allow for the highest level of aircraft performance under adverse conditions [15]. In addition to the modification of the aircraft control laws the estimated parameter information can be used to track the health of the system. As parameters diverge from the nominal value this divergence can be used to understand failure progressions and help to detect or predict failures in the aircraft.

The primary focus of this work will be in the area of real-time parameter estimation. The parameter estimation method that will be investigated will be based on the nonlinear sliding mode control architecture. Understanding of sliding mode control along with how it can be used as a parameter estimation technique will be the first phase of the study, followed by application to nonlinear aircraft models and simulations. The work is intended to prove the theory and supply initial results for a novel parameter estimation technique using sliding mode theory. Results will be compared with state-of-the art model reference parameter estimation techniques. The inputs to the aerodynamic model will be both sin inputs and standard doublet maneuvers. The majority of the work will be completed in MATLAB and Simulink utilizing nonlinear aircraft models.

A description of the estimation development objectives are listed below:

1. Analyze the proposed estimation technique on first order, second order, and nonlinear proof of concept models.

2. Develop a real-time sliding mode aerodynamic coefficient estimator for a Lockheed Martin F-16 nonlinear aircraft model, specifically the pitching moment coefficient C_m, the lifting force coefficient C_L, and the drag force coefficient C_D.

   - Derive the estimation equations
   - Simulate the estimators
   - Analyze the performance of the estimators

3. Compare the results of the sliding mode estimator with a common model reference estimation technique.

Chapter 2

System Model Definition

Multiple different systems are used to analyze the capability and performance of the PID algorithm, a variety of linear, nonlinear, and second order systems have been constructed and developed to test the identification methodology and implementation. Outlined in Section 2.1 are the basic equations that have been used to prove out and rapidly develop the estimator prior to implementation on the aircraft model. Section 2.2 then shows the development of the nonlinear F-16 aircraft model, outlining the rigid body and Euler kinematic equations, stability axis coordinate system, stability derivatives, and the aircraft trim conditions.

2.1 Proof of Concept

2.1.1 First Order Model

The first model that was used for PID algorithm development is shown in Equation 2.1, this is a first order linear system with unknown parameter $\hat{a}$ and input u. A PID algorithm has been developed and the derivation for this system can be found in Section 3.3.1 and simulation and analysis results can be seen in Section 4.1.1.

$$\dot{x} = -\hat{a}x + u \tag{2.1}$$

2.1.2 First Order Nonlinear Model

After PID was successfully implemented for the first order model, a PID algorithm was developed for the first order nonlinear system of equations defined by Equation 2.2. This system of equations includes a nonlinear $sin(y)$ function along with multiple states x and y. The parameter that is estimated in this system of equations is $\hat{a}$, the input to the system is u, B and C are both constants, and $\sin(\omega t)$ is a forcing function input.

$$\begin{aligned} \dot{x} &= -\hat{a}\sin(y)x + Bu \\ \dot{y} &= -Cyx + \sin(\omega t) \end{aligned} \tag{2.2}$$

A PID algorithm has been developed and the derivation for the first order non linear system can be found in Section 3.3.2 and simulation and analysis results can be seen in Section 4.1.2.

2.1.3 Second Order Model

For the successful implementation of this parameter estimator on the aircraft system, multiple parameters need to be estimated simultaneously, a PID algorithm that estimates two parameters was developed for the second order system defined by Equation 2.3. The parameters that are estimated in this system are $\hat{a}_1$ and $\hat{a}_2$, the input to the system is u.

$$\ddot{x} = -\hat{a}_2\dot{x} - \hat{a}_1 x + u \tag{2.3}$$

A PID algorithm has been developed and the derivation for the second order system can be found in Section 3.3.3 and simulation and analysis results can be seen in Section 4.1.3. The derivation and results are for both the $\hat{a}_1$ and $\hat{a}_2$ parameters.

2.1.4 Second Order Nonlinear Model

Multiple parameters need to be estimates simultaneously on a nonlinear system, a PID algorithm that estimated multiple parameters was developed for the second order nonlinear system defined by Equation 2.4. The parameters that are estimated in this system are $\hat{a}_1$ and $\hat{a}_2$, the input to the system is u.

$$\ddot{x} = -\hat{a}_2\dot{x} - \hat{a}_1 x^2 + u \tag{2.4}$$

A PID algorithm has been developed and the derivation for the second order nonlinear system can be found in Section 3.3.4 and simulation and analysis results can be seen in Section 4.1.4. The derivation and results are for both the $\hat{a}_1$ and $\hat{a}_2$ parameters.

2.2 Aircraft Model

2.2.1 Nonlinear Model

Rigid Body Equations

A nonlinear model was developed for three-dimensional simulation of the Lockheed Martin F-16 aircraft, the model will be used for the PID development as well as analysis and verification of the estimators capability. The aircraft equations of motion were derived from first principles and simulated in the MATLAB and Simulink environment. Table 2.1 displays a list and definition of variables used in the derivation of the nonlinear aircraft model. Newton's second law is used to derive the rigid body equations of motion, the law states that the time rate of change of the momentum of the body is equal to the summation of all the esternal forces and the time rate of change of angular momentum is equal to the summation of the external moments acting on the body. Newton's second law is reporesented by the following vector equations [16].

$$\begin{aligned} \sum \vec{\mathbf{F}} &= \tfrac{d}{dt}\left(m\vec{\mathbf{v}}\right) \\ \sum \vec{\mathbf{M}} &= \tfrac{d}{dt}\vec{\mathbf{H}} \end{aligned} \tag{2.5}$$

In Equation 2.5, $\vec{\mathbf{F}}$ is the net vector force on the rigid body, m is the mass of the aircraft, $\vec{\mathbf{v}}$ is the velocity, $\vec{\mathbf{M}}$ is the net moment vector of the system, and $\vec{\mathbf{H}}$ is the angular momentum, referenced to the coordinate frame. The scalar force and moment equations of Equation 2.5 are expressed below in Equation 2.6.

$$\begin{array}{lll} F_x = \frac{d}{dt}(mu) & F_y = \frac{d}{dt}(mv) & F_z = \frac{d}{dt}(mw) \\ L = \frac{d}{dt}H_x & M = \frac{d}{dt}H_y & N = \frac{d}{dt}H_z \end{array} \tag{2.6}$$

The force components of the x, y, and z axes are represented by F_x, F_y, and F_z; velocity components are also represented as u, v, and w, respectively. The moment

Table 2.1: Nomenclature for the Forces, Moments, and Rates for the Aircraft Fixed Body Axis System

Parameter Name	Roll Axis x_b	Pitch Axis y_b	Yaw Axis z_b
Aerodynamic force components	X	Y	Z
Aerodynamic moment components	L	M	N
Angular rates	p	q	r
Velocity components	u	v	w
Moment of inertia	I_{xx}	I_{yy}	I_{zz}
Products of inertia	I_{yz}	I_{xz}	I_{xy}

components along the x, y, and z axes are represented by L, M, and N; moment of momentum components are also represented as H_x, H_y, and H_z, respectively.

When each of the individual mass elements of the aircraft are considered, Newton's second law can be written as Equation 2.7 where $\vec{\mathbf{v}}_c$ is the velocity of the center of mass of the airplane and ${}^{d\vec{\mathbf{r}}}/_{dt}$ is the velocity of the element relative to the center of mass, δm.

$$\sum \delta\vec{\mathbf{F}} = \vec{\mathbf{F}} = \frac{\mathrm{d}}{\mathrm{dt}} \sum \left(\vec{\mathbf{v}}_c + \tfrac{d\vec{\mathbf{r}}}{dt}\right)\delta m \tag{2.7}$$

The assumption will be made that the mass of the aircraft remains constant and since $\vec{\mathbf{r}}$ is measured from the center of mass, the summation $\sum \vec{\mathbf{r}}\delta m$ is equal to zero and Equation 2.7 can be reduced to Equation 2.8 which relates the external force on the airplane to the motion of the vehicle's center of mass.

$$\vec{\mathbf{F}} = m\tfrac{d\vec{\mathbf{v}}_c}{dt} \tag{2.8}$$

In a similar fashion, the moment equation referring to moving center of mass for a differential element can be derived, for the differential element of mass , δm, the moment equation can be written as Equation 2.9.

$$\delta\vec{\mathbf{M}} = \frac{d}{dt}\delta\vec{\mathbf{H}} = \frac{d}{dt}\left(\vec{\mathbf{r}} \times \vec{\mathbf{v}}\right)\delta m \tag{2.9}$$

Where the velocity of mass element, $\vec{\mathbf{v}}$, in Equation 2.9 is equal to the expression in Equation 2.10.

$$\vec{\mathbf{v}} = \vec{\mathbf{v}}_c + \vec{\omega} \times \vec{\mathbf{r}} \tag{2.10}$$

Now it is assumed that $\vec{\mathbf{v}}_c$ is constant and substituting Equation 2.10 into Equation 2.9 and re-arranging results in the moment equation defined in Equation 2.11.

$$\vec{\mathbf{H}} = \sum \vec{\mathbf{r}} \delta m \times \vec{\mathbf{v}}_c + \sum \left[\vec{\mathbf{r}} \times (\vec{\omega} \times \vec{\mathbf{r}}) \right] \delta m \tag{2.11}$$

Equation 2.11 can be written in matrix form as shown in Equation 2.12 where p, q, and r are the angular rates about the x, y, and z axes, respectively.

$$\vec{\mathrm{H}} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2.12}$$

The matrix in Equation 2.12 is the inertia tensor for the aircraft, each of the different components of the matrix are defined in Equation 2.13.

$$\begin{aligned} I_{xx} &= \int\int\int (y^2 + z^2)\, \delta m \quad I_{xy} = \int\int\int xy \delta m \\ I_{yy} &= \int\int\int (x^2 + z^2)\, \delta m \quad I_{xz} = \int\int\int xz \delta m \\ I_{zz} &= \int\int\int (x^2 + y^2)\, \delta m \quad I_{yz} = \int\int\int yz \delta m \end{aligned} \tag{2.13}$$

The terms I_{xx}, I_{yy}, and I_{zz} are the mass moments of inertia of the body about the x, y, and z axes, respectively. The other three terms defined in Equation 2.13 are called the products of inertia. Both the moments and products of inertia are defined based on the shape of the aircraft and the manner in which the mass is distributed. In a fixed reference frame, if the aircraft rotates these terms will change. The axis system is fixed to the aircraft to assume the mass moments of inertia remain constant for a given aircraft. The derived force and moment equations of motion are defined in Equations 2.14 and 2.15 for the aircraft fixed axis system.

$$\vec{\mathbf{F}} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} m\left(\dot{u} + qw - rv\right) \\ m\left(\dot{v} + ru - pw\right) \\ m\left(\dot{w} + pv - qu\right) \end{bmatrix} = \vec{\mathbf{W}}_{ref} + \vec{\mathbf{F}}_{aerodynamic} + \vec{\mathbf{F}}_{thrust} \tag{2.14}$$

$$\vec{\mathbf{M}} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \tag{2.15}$$

$$\begin{pmatrix} qr\left(I_{yy} - I_{zz}\right) + \left(q^2 + r^2\right) I_{xy} - prI_{xy} + pqI_{xz} \\ pr\left(I_{zz} - I_{xx}\right) + \left(r^2 + p^2\right) I_{xz} - pqI_{yz} + qrI_{xy} \\ pq\left(I_{xx} - I_{yy}\right) + \left(p^2 + q^2\right) I_{xy} - qrI_{xz} + prI_{yz} \end{pmatrix} + \vec{\mathbf{M}}_{external}$$

The equations of motion that are defined in Equations 2.14 and 2.15 have been derived for a generic aircraft, the F-16 aircraft mass and inertia tensor values will be used for accurate modeling of the F-16 aircraft. The components of the force and moment acting on the airplane are composed of aerodynamic, gravitational, and propulsive contributions; these components are covered in detail in the Stability Derivative Equations, Section 2.2.1.

Euler Kinematic Equations

The equations of motion derived in Section 2.2.1 are for the body fixed axis system, these equations can not be used to describe the position and orientation of the aircraft relative to a moving body axis frame. In order to track the orientation of the aircraft to the earth fixed axis system a standard method for angular transformations will be used and described in this section. The orientation relative to the earth fixed axis system can be described by three consecutive rotations, these angular rotations are called the Euler angles. The Euler angle rotation is a standard sequence of applying rotations for aerospace axis references. The rotations are applied about the primary, secondary, and tertiary axis (x_b, y_b, z_b) in that specific order.

Equation 2.16 defines the transformation matrix for each of the three rotations and Equation 2.17 shows how the rotation matrices are applied in order to transform the rotations from the earth fixed axis system to the body fixed axis system and vice-verse, respectively.

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

$$R_2 = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\theta \end{bmatrix} \tag{2.16}$$

$$R_3 = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{E\to V} = R_1 \cdot R_2 \cdot R_3 \tag{2.17}$$

$$T_{V\to E} = R_3 \cdot R_2 \cdot R_1$$

The transformation equations in matrix form for converting from the earth fixed axis system to the body fixed axis system (and vice-versa) are shown in Equations 2.18 and 2.19. Singularities in the matrix exist in Equation 2.19 when the pitch angle θ approaches 90 degrees. This singularity can be avoided b using a quaternion transformation, the quaternion transformation is not necessary for this application because the pitch angle during estimation maneuvers will never approach 90 degrees.

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{2.18}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2.19}$$

Stability Axis Coordinate System

The stability axis coordinate system provides the foundation for the nonlinear equations of motion for the aircraft. The stability axis system is based on the three aircraft states; the true velocity, V_T, the angle of attack, α, and the the sideslip angle, β. The aircraft's true velocity is the magnitude of the body fixed axis velocity components, the angle of attack is defined as the pitch angle of the aircraft relative to the oncoming wind, and the sideslip angle is also measured relative to the oncoming wind.

Equations 2.20 provides the transformation from the body fixed axis system to the stability axis system and Equation 2.21 provides the transformation from the stability axis to the body fixed axis system.

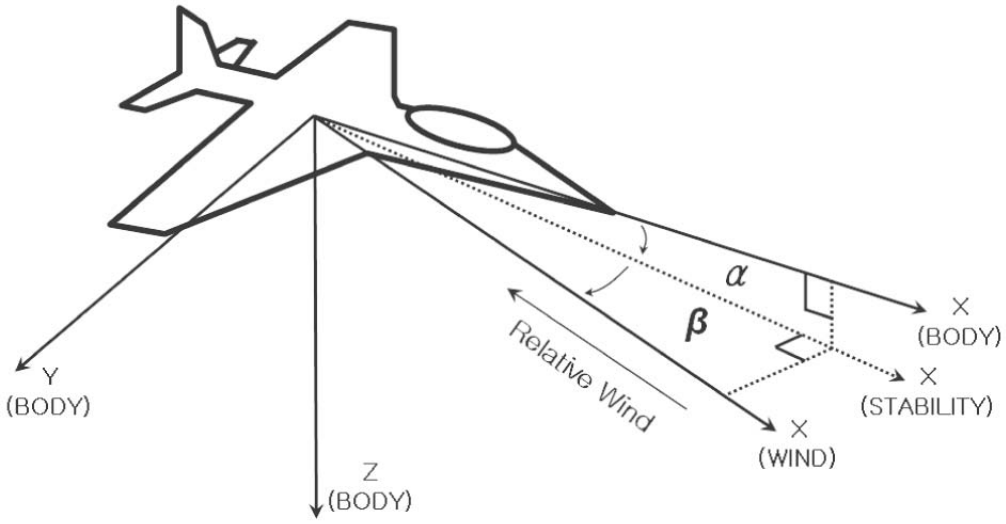

Figure 2.1: Illustration of Stability Axis and Aircraft Body Axis [17]

$$\begin{aligned} \alpha &= \arctan \tfrac{w}{u} \\ \beta &= \arcsin \tfrac{u}{V_T} \\ V_T &= \sqrt{u^2 + v^2 + w^2} \end{aligned} \tag{2.20}$$

$$\begin{aligned} u &= V_T \cos\alpha \cos\beta \\ v &= V_T \sin\beta \\ w &= V_T \sin\alpha \cos\beta \end{aligned} \tag{2.21}$$

The stability axis force equations are defined in Equation 2.22, parameters DOM, YOM, and LOM are defined by Equation 2.23. The variables D, T, Y, and L in

equation 2.23 represent the drag, thrust, side, and lift forces acting on the aircraft. The drag, side, and lift forces are assumed to act along the stability axis while thrust acts along the aircraft body axis, x_b.

$$\begin{aligned}
&\dot{\alpha} = q - (p\cos\alpha + r\sin\alpha)\tan\beta - \tfrac{LOM}{V_T\cos\beta} + \\
&\tfrac{g}{V_T\cos\beta}(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) \\
&\dot{\beta} = p\sin\alpha - r\cos\alpha + \tfrac{1}{V_T}(YOM\cos\beta + DOM\sin\beta) + \\
&\tfrac{g}{V_T}(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha) \\
&\dot{V}_T = YOM\sin\beta - DOM\cos\beta + \\
&g\left[(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\cos\beta + \cos\theta\sin\phi\sin\beta\right] \\
&DOM = \tfrac{D - T\cos\alpha}{m}; \quad YOM = \tfrac{Y}{m}; \quad LOM = \tfrac{L + T\sin\alpha}{m} \\
&D = \bar{q}SC_D; \qquad Y = \bar{q}SC_Y; \qquad L = \bar{q}SC_L
\end{aligned} \tag{2.22, 2.23}$$

Stability Derivative Equations

The external forces and moments that were referenced in the previous section are composed of aerodynamic, gravitational, and thrust contributions. Equation 2.24 gives the components of the aircraft weight force, where θ is the pitch angle and ϕ is the bank angle of the aircraft.

$$\mathbf{W} = mg \begin{bmatrix} \sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix} \tag{2.24}$$

Equations 2.25 and 2.26 define the force and moment vectors, where $\bar{q}$ is dynamic pressure defined in equation 2.27, S is the wing span reference area, $\bar{c}$ is the wing mean aerodynamic chord, and b is the wing span. The force and moment coefficients are shown as a $C_{()}$ with a subscript, these are determined by the summation of contributions to each term.

$$\mathrm{F}_{aerodynamic} = \begin{bmatrix} Drag \\ Side \\ Lift \end{bmatrix} = \begin{bmatrix} \bar{q}SC_D \\ \bar{q}SC_Y \\ \bar{q}SC_L \end{bmatrix} \tag{2.25}$$

$$\mathrm{M}_{external} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \bar{q}SbC_l \\ \bar{q}S\bar{c}C_m \\ \bar{q}SbC_n \end{bmatrix} \tag{2.26}$$

$$\bar{q} = \tfrac{1}{2}\rho(V_T)^2 \tag{2.27}$$

Equation 2.28 gives the components of force and moment coefficients, the coefficients are defined by nonlinear lookup tables based on the F-16 aircraft states and configuration. Equations 2.29 and 2.30 represent linear approximations of the nonlinear lookup tables, these are incorporated into the modeling of the aircraft but are not used during parameter estimation. The variables δ_a, δ_e, δ_f, and δ_r are the control sufrace deflections corresponding to the aileron, elevator, flap, and rudder control surfaces.

$$\begin{aligned}
&\text{Longitudinal Axis :} \\
&C_D = f(\alpha, \delta_e) \\
&C_L = f(\alpha, \dot{\alpha}, q_b, \delta_e, \delta f, V_T) \\
&C_m = f(\alpha, \dot{\alpha}, q_b, \delta_e, V_T) \\
&\\
&\text{Lateral Directional Axis :} \\
&C_Y = f(\beta, \delta_r) \\
&C_l = f(\alpha, \beta, p_b, r_b, \delta_a, \delta_r, V_T) \\
&C_n = f(\alpha, \beta, p_b, r_b, \delta_a, \delta_r, V_T)
\end{aligned} \tag{2.28}$$

$$\begin{aligned}
\mathrm{C_D} &= \mathrm{C_{D_0}} + C_{D_\alpha}\alpha + \tfrac{c}{2V_T}\left(C_{D_q}q + C_{D_{\dot{\alpha}}}\dot{\alpha}\right) + C_{D_{\delta_e}}\delta_e + C_{D_{\delta_f}}\delta_f \\
C_Y &= C_{Y_0} + C_{Y_\beta}\beta + \tfrac{b}{2V_T}\left(C_{Y_p}p + C_{Y_r}r\right) + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r \\
C_L &= C_{L_0} + C_{L_\alpha}\alpha + \tfrac{c}{2V_T}\left(C_{L_q}q + C_{L_{\dot{\alpha}}}\dot{\alpha}\right) + C_{L_{\delta_e}}\delta_e + C_{L_{\delta_f}}\delta_f
\end{aligned} \tag{2.29}$$

$$\begin{aligned}
C_l &= C_{l_0} + C_{l_\beta}\beta + \tfrac{b}{2V_T}\left(C_{l_p}p + C_{l_r}r\right) + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r \\
C_m &= C_{m_0} + C_{m_\alpha}\alpha + \tfrac{c}{2V_T}\left(C_{m_q}q + C_{m_{\dot{\alpha}}}\dot{\alpha}\right) + C_{m_{\delta_e}}\delta_e + C_{m_{\delta f}}\delta_f \\
C_n &= C_{n_0} + C_{n_\beta}\beta + \tfrac{b}{2V_T}\left(C_{n_p}p + C_{n_r}r\right) + C_{n_{\delta a}}\delta_a + C_{n_{\delta_r}}\delta_r
\end{aligned} \tag{2.30}$$

2.2.2 Aircraft Trim Conditions

Trim is defined as the lack of rotation about an aircraft's center of gravity when not performing flight maneuvers [16]. The trim conditions for the F-16 aircraft are given in this section, a MATLAB script was used to identify the proper trim conditions for the aircraft. The trim condition identification routine was executed in the following manner; first a Simulink model of the F-16 aircraft was run with an initial guess of trim conditions, after this initial guess the aircraft states are varied slightly and the model in run after each change in parameter. This process stops once the trim index (defined in Equation 2.31) is within a specified tolerance.

$$\text{trim\_index} = \sqrt{\dot{V}_T^2 + \dot{\alpha}^2 + \dot{\beta}^2 + \dot{p}^2 + \dot{q}^2 + \dot{r}^2} \tag{2.31}$$

2.2.3 F-16 Trim Specifications

The F-16 model defined in Section 2.2 was trimmed at an altitude of $20,000 ft$ and a desired airspeed of $675 ft/sec$. The values that have been defined as the trim condition for all testing are defined in Table 2.2. These values will be used as the initial starting point, prior to any command input or parameter estimation routine.

Table 2.2: F-16 Trim Conditions

Altitude (ft)	2.0000e+04
Mach	6.5130e-001
V_T (ft/s)	6.7527e+002
α (deg)	3.5970e+000
β (deg)	0
ϕ (deg)	0
θ (deg)	3.5970e+000
ψ (deg)	0

2.2.4 NASA Dryden Wind Gust Model

NASA Dryden has developed a time-based wind gust model that will be used to simulate winds in all three axis of the nonlinear aircraft model. The equations that are implemented for this simulation are described below, the magnitude of the wind gust in any direction is defined by Equation 2.32. There are three parameters that need to be defined for each of the three axis of wind, the first parameter is the U_{wind} which is the end magnitude of the wind gust, in ft/sec. The second parameter that must be defined is the U_wind_d which is the time to reach the peak wind from when the wind gust begins. The final parameter that needs to be defined is the U_wind_Td which is the time delay for the wind gust that will be applied. An example output of the wind model is shown in Figure 2.2, for this example U_{wind} was defined as 20 ft/sec, U_wind_d was 6 seconds, and U_wind_Td was 2 seconds. The wind gust generator is modeled in Simulink and can be applied in any of the thee axis of the aircraft.

$$\begin{aligned} &\frac{|U_{wind}|}{2} \cdot \left[1 - \frac{\cos \pi t}{U\_wind\_d}\right] \quad \text{for} \quad U \leq U\_wind\_d \\ &|U_{wind}| \qquad\qquad\qquad\qquad \text{for} \quad U > U\_wind\_d \end{aligned} \tag{2.32}$$

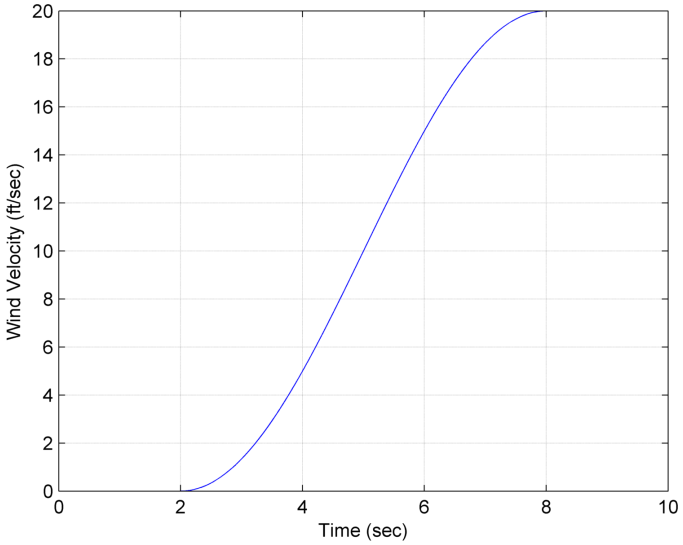

Figure 2.2: Plot of Wind-Gust Along x_b Axis Over Time

Chapter 3

Real-Time Parameter Identification

3.1 Model Reference Adaptive Control

The idea behind Model Reference Adaptive Control (MRAC) is to create a closed loop controller with parameters that can be updated to change the response of the system. The output of the system is compared to a desired response from a reference model and the control parameters are updated based on this error. The overarching objective is parameter convergence that allows the plant response to match that of the reference model response.

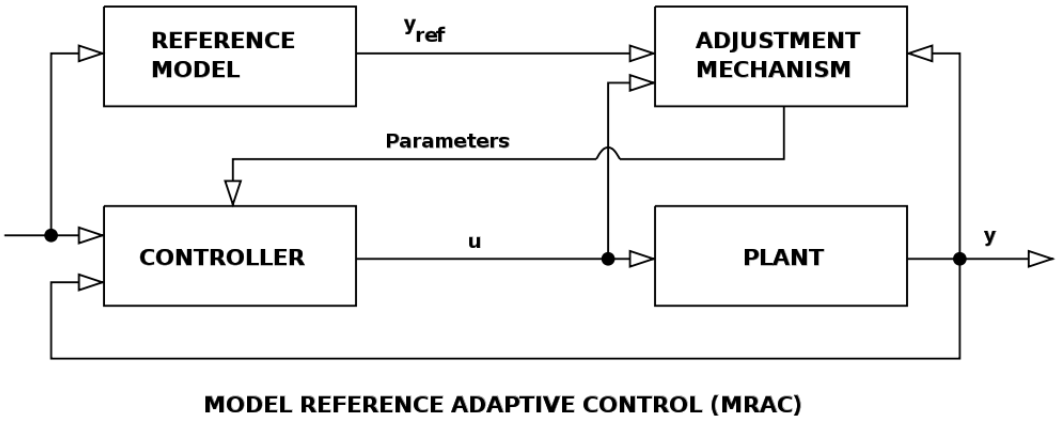

Figure 3.1: Model Reference Adaptive Control Overview

Figure 3.1 shows the block diagram of a MRAC system, the reference model block is the block that creates the desired states, y_{ref}, the controller block is the control system for the plant that can have internal parameters updated, the adjustment mechanism

is the block that creates updated controller parameters from the plant and reference model error, and the plant is the system that is being controlled. This type of parameter updating process can be used to estimate parameters of a system if the basic system model is understood. Instead of feeding the updated parameters into a control system, they will be fed into the model and then compared to the actual system plant. The parameters that are updated will then match the actual parameters in the plant and will track if they are to change over time. The parameter identification derivation based on the MRAC technique is shown in Section 3.1.1.

3.1.1 Aircraft MRAC PID Derivation

This section will show the MRAC derivation for the F-16 aircraft model shown in Section 2.2. The first step in the MRAC PID estimator development process is to define the difference between the plant output and the reference model output. This difference will be used as a metric to compare both systems and determine if they are acting the same. For the aircraft model this tracking error e is defined in Equation 3.1.

$$e = (p - p_m) + (q - q_m) + (r - r_m) \tag{3.1}$$

From this error equation, a cost function is created for the aircraft parameter that will be estimated. This cost function is shown as Equation 3.2.

$$J(\hat{C}_{m_q}) = \tfrac{1}{2}\left[\left(\hat{p}^2 - p_m^2\right) + \left(\hat{q}^2 - q_m^2\right) + \left(\hat{r}^2 - r_m^2\right)\right] \tag{3.2}$$

To find out how the parameter $\hat{C}_{m_q}$ will be updated, an equation for the change in the parameter is developed where the goal is to minimize the cost function. The cost function will be minimized because a differential equation (Equation 3.3) is set up to move in the direction of the negative gradient of the cost function shown in Equation 3.2.

$$\dot{\hat{C}}_{m_q} = -\lambda\left[\left(\hat{p}\dot{\hat{p}} - p_m\dot{p}_m\right) + \left(\hat{q}\dot{\hat{q}} - q_m\dot{q}_m\right) + \left(\hat{r}\dot{\hat{r}} - r_m\dot{r}_m\right)\right] \tag{3.3}$$

Equation 3.3 is programmed into the Matlab and Simulink simulation and compared to the results of the newly developed parameter estimators. The results for this model reference estimator can be found in Section 4.2.2.

3.2 Sliding Mode Controller

This description and explanation of sliding mode control and the example derivation of control law are given in the following section as provided by Slotine and Li [18]. A similar approach is used in the development of the parameter estimation routine used for nonlinear aircraft parameter identification. The state tracking and robust convergence attributed to the sliding mode controller make it an exceptional candidate for parameter estimation, in this section a brief background on sliding mode control is presented along with an explanation of how sliding mode control theory will be applied to parameter estimation.

3.2.1 Sliding Mode Control

Sliding mode control is a form of robust control based on the concept that controlling a 1^{st}-order system is easier than controlling a n^{th}-order system. This notion is applicable to both nonlinear and uncertain systems. Simplification in the representation and notation of the problem allows for n^{th}-order systems to be represented by 1^{st} order systems where it can be shown that desired controller performance can be achieved, this controller performance comes with the cost of increased controller complexity and activity. An explanation of this transformation and control input derivation is shown below considering a general system given by Equation 3.4. The variable $x^{(n)}$ is the scalar output, $\mathbf{x}$ is the state-variable (given by Equation 3.5) and u is the control input.

$$x^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u \tag{3.4}$$

$$\mathbf{x} = \begin{bmatrix} x & \dot{x} & \dots & x^{(n-1)} \end{bmatrix}^T \tag{3.5}$$

Sliding mode controllers are designed to be state tracking control schemes, in this example the goal of the controller is for system state variable $\mathbf{x}$ to track $\mathbf{x}_\text{d}$. The tracking error is represented by $\tilde{\mathbf{x}}$, which is defined in Equation 3.6.

$$\tilde{\mathrm{x}} = \mathrm{x} - \mathrm{x}_d = \begin{bmatrix} \tilde{x} & \dot{\tilde{x}} & \dots & \tilde{x}^{(n-1)} \end{bmatrix}^T \tag{3.6}$$

The time-varying surface, s(t), which is better know as the sliding surface, is defined by Equation 3.7. the variable λ is a strictly positive constant.

$$s(\mathbf{x};t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \tilde{x} \tag{3.7}$$

Equation 3.8 shows the sliding surface for a second order system, where $n = 2$.

$$s = \dot{\tilde{x}} + \lambda\tilde{x} \tag{3.8}$$

The derivative of the sliding surface, $\dot{s}$, is given by Equation 3.9.

$$\dot{s} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}} \tag{3.9}$$

Setting the derivative of the sliding surface equal to zero ensures that once the sliding surface has been reached, the system remains there and the error will reduce to zero. Applying this condition, the control input can then be determined. For the case of the 2^{nd}-order system given by Equation 3.10, the sliding surface can be determined using the steps from Equation 3.7 through Equation 3.9.

$$\ddot{x} = f(x) + u \tag{3.10}$$

The sliding surface is the same as in Equation 3.9, as $n = 2$ in this example. Substituting Equation 3.10 into Equation 3.9 and solving for zero, yields the control input u. These steps are shown in Equation 3.11.

$$\begin{aligned} \dot{s} &= f(x) + u - \ddot{x}_d + \lambda\dot{\tilde{x}} = 0 \\ u &= -f(x) + \ddot{x}_d - \lambda\dot{\tilde{x}} \end{aligned} \tag{3.11}$$

3.2.2 Sliding Mode Parameter Estimation

Using the sliding mode control process, shown in Section 3.2.1, as a foundation, a parameter estimator can be developed that can be used for both positive and negative parameters and can guarantee stability and convergence. This new type of estimator is highly desirable for aircraft parameter estimation. This section describes the newly created estimator development process, the estimator derivations are shown in Section

3.3 and 3.4, and the simulation results for these estimators can be found in Section 4. An overview diagram of the estimation process is shown in Figure 3.2. The reference model will be run in the Simulink environment for testing but when the estimator is deployed it will use signals from the actual aircraft, the estimated model will run the parameters that are identified in the parameter estimator block, and the parameter estimator will run taking into account the system inputs and both model and reference states. The resulting output of this system will be parameters that have converged upon the actual parameters inside of the reference system.

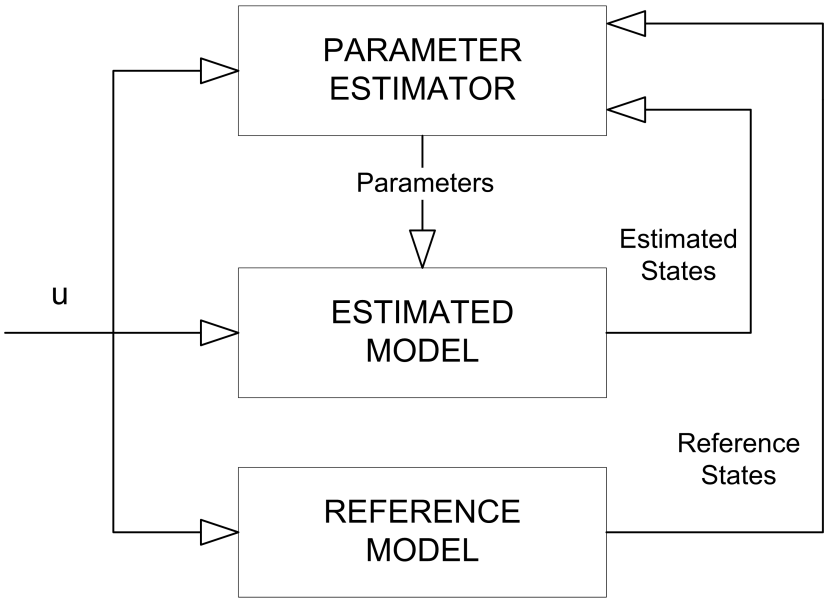

Figure 3.2: Parameter Estimation Overview

The sliding mode control development shown in Section 3.2.1 is focused primarily on the calculation and identification of the control input, u. This goal changes significantly when a parameter estimator is developed, the primary goal in the estimator development is the calculation and identification of one of the system parameters. So instead of developing an equation for u, like the one shown in Equation 3.11, an equation for one of the system parameters will be developed instead. The main change in the sliding mode control process is the development of the sliding surface, when developing the sliding surface for the parameter estimator, the parameter that needs to be estimated must be included in the sliding surface. This allows the derivative of the parameter to be solved for further along in the development process. Using the example system of equations

from Section 3.2.1, Equation 3.4 is re-written here as Equation 3.12.

$$x^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u \tag{3.12}$$

Equation 3.12 will be used as the system equation for an example parameter estimator development, it is assumed that the function $f(\mathbf{x})$ has a parameter a that will be estimated. The sliding surface for a second order system that will be developed for the parameter estimation routine is shown as Equation 3.13.

$$s = \dot{\tilde{x}} + \lambda\tilde{x} \tag{3.13}$$

The derivative of the sliding surface, $\dot{s}$, is then given by Equation 3.14, an additional $[K(\dot{\mathrm{x}} - \dot{\mathrm{x}}_d) + \eta]\,\dot{\hat{a}}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined to provide Lyapunov stability and guarantee convergence.

$$\dot{s} = \ddot{x} - \ddot{x}_d + \lambda\dot{\tilde{x}} - [K(\dot{\mathrm{x}} - \dot{\mathrm{x}}_d) + \eta]\,\dot{\hat{a}} \tag{3.14}$$

Setting the derivative of the sliding surface equal to zero ensures that once the sliding surface has been reached, the system remains there and the error will be reduced to zero. Applying this condition, the unknown parameter can then be estimated. The η variable has been added to prevent a divide by zero condition during numerical simulation, this is an extremely small positive value. For the case of the 2^{nd}-order system given by Equation 3.10 and the derivative of the sliding surface is given in Equation 3.13. Substituting Equation 3.10 into Equation 3.13 and solving for zero, yields the unknown parameter, $\dot{\hat{a}}$. These steps are shown in Equations 3.15 and Equation 3.16.

$$\dot{s} = f(x) + u - \ddot{x}_d + \lambda\dot{\tilde{x}} - [K(\dot{\mathrm{x}} - \dot{\mathrm{x}}_d) + \eta]\,\dot{\hat{a}} = 0 \tag{3.15}$$

$$\dot{\hat{a}} = \frac{-f(x) - u + \ddot{x}_d - \lambda\dot{\tilde{x}}}{K(\dot{\mathrm{x}} - \dot{\mathrm{x}}_d) + \eta} \tag{3.16}$$

The variable η is used to prevent a divide by zero condition, this is a very small positive constant. The variable K will be defined by an equation that will guarantee Lyapunov stability for parameter variance within and upper and lower bounds. Lyapunov stability is defined in the next section (Section 3.2.3). Equation 3.16 is the equation that would be

used to estimate the unknown parameter, this process will be used to derive estimation equations for all of the systems defined in Section 2.

3.2.3 Lyapunov Theorem for Global Stability

One of the main advantages of the sliding mode estimator that is proposed in this paper is the ability to guarantee stability and robustness; this will ensure proper convergence to the parameter that is being estimated. We are interested in designing a sliding parameter estimator such that a stable sliding mode is generated as well as asymptotic stability for the estimator. The method described in this section is based on the selection of a Lyapunov function [19]. The estimator sliding function should be chosen such that the candidate Lyapunov function satisfies Lyapunov stability criteria. Lyapunov global stability is stated in the following theorem.

Assume there exists a scalar function V of S, with continuous first order derivatives such that

- $V(S)$ is positive definite
- $\frac{dV(S)}{dt}$ is negative definate
- $V(S) -> \infty$ as $\|S\| -> \infty$

Then the equilibrium at the $S = 0$ is globally and asymptotically stable [18].

3.3 Proof of Concept Parameter Identification Development

PID Derivations are shown in this section for each of the proof of concept system models that have been defined in Chapter 2. The novel parameter identification method that has been developed is based upon the sliding mode control system development shown in Section 3.2, the parameter estimator is developed to guarantee a robust and stable convergence to the estimated value by following a traditional sliding surface and Lyapunov function. For each of the proof of concept system models a step by step development process will be shown, with the end result being an estimation equation for each of the parameters that are being estimated.

3.3.1 First Order Model Implementation

Parameter identification derivation for the first order system described in Section 2.1.1 is show, consider the system equation is defined as Equation 3.17 with uncertain parameter $\hat{a}$.

$$\dot{x} = -\hat{a}x + u \tag{3.17}$$

The candidate Lyapunov function is defined by Equation 3.18

$$V = \tfrac{1}{2}s^2 \geq 0 \tag{3.18}$$

The Lyapunov function described in Equation 3.18 is positive semi-definite for all time, the sliding surface is then defined as Equation 3.19. x_m and $\dot{x}_m$ are the measured states from the reference system.

$$s = \tfrac{1}{2}\left(x^2 - x_m^2\right) + \lambda \int \tfrac{1}{2}\left(x^2 - x_m^2\right) \tag{3.19}$$

To ensure no movement off of the sliding surface set $\dot{s} = 0$, shown in Equation 3.20. An additional $\left[K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined to provide Lyapunov stability and guarantee convergence. The η parameter in the sliding surface derivative is added to prevent a divide by zero condition in the numerical simulation.

$$\dot{s} = \left(x\dot{x} - x_m\dot{x}_m\right) + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) - \left[K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}} = 0 \tag{3.20}$$

Substitute equation 3.17 into Equation 3.20 yields Equation 3.21

$$\dot{s} = -\hat{a}x^2 + ux - x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) - \left[K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}} = 0 \tag{3.21}$$

Re-arranging and solving for $\dot{\hat{a}}$ yields Equation 3.22.

$$\dot{\hat{a}} = \frac{1}{K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta}\left[ux - \hat{a}x^2 - x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right)\right] \tag{3.22}$$

Equation 3.22 defines the equation that will be programmed into MATLAB and Simulink to determine $\hat{a}$. To satisfy the Lyapunov theorem of global stability as defined in Section 3.2.3, gains need to be chosen so that Equation 3.23 is true, ensuring stability and convergence.

$$\dot{V} = s\dot{s} < 0 \tag{3.23}$$

A more convenient choice still satisfying Equation 3.23 is Equation 3.24.

$$s\dot{s} \leq -\left(K + \eta\right)|s| \tag{3.24}$$

Where K and η are strictly positive values. Consider the actual system defined by Equation 3.25.

$$\dot{x} = -ax + u \tag{3.25}$$

Substituting Equation 3.25 into Equation 3.20:

$$\dot{s} = -ax^2 + ux - x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) - \left[K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}} = 0 \tag{3.26}$$

Now substituting Equation 3.22 into Equation 3.26:

$$\dot{s} = \hat{a}x^2 - ax^2 \tag{3.27}$$

Substituting Equation 3.27 into Equation 3.24:

$$K|s| \leq \left[\left(a - \hat{a}\right)x^2\right]s - \eta|s| \tag{3.28}$$

The parameter a is unknown but assumed to be bounded and defined by the Equation 3.29.

$$a_{low} \leq a \leq a_{upp} \tag{3.29}$$

Assume the parameter a is the geometric mean of the bounds defined in Equation 3.29

$$a = \sqrt{a_{low}a_{upp}} \tag{3.30}$$

Applying Equation 3.30 to Equation 3.29 to create a ratio of a and $\hat{a}$ results in Equation 3.31.

$$\alpha \leq \frac{a}{\hat{a}} \leq \alpha^{-1} \tag{3.31}$$

Where:

$$\alpha = \sqrt{a_{upp}/a_{low}} \tag{3.32}$$

Equation 3.28 can be re-written as Equation 3.33.

$$K|s| \leq \left[\hat{a}\left(\frac{a}{\hat{a}} - 1\right)x^2\right] s - \eta|s| \tag{3.33}$$

Comparing the right side of the equality constraint of Equation 3.31 which is taken to be the upper bound limit and the most conservative estimate for the unknown parameter to Equation 3.33 results in Equation 3.34.

$$K|s| \leq \left[\hat{a}\left(\alpha^{-1} - 1\right)x^2\right] s - \eta|s| \tag{3.34}$$

Equation 3.34 can be satisfied by Equation 3.35.

$$K|s| \leq \left|\hat{a}\left(\alpha^{-1} - 1\right)x^2\right| |s| - \eta|s| \tag{3.35}$$

The resulting value for the gain K using the equality constraint of Equation 3.35 becomes Equation 3.36.

$$K = \left|\hat{a}\left(\alpha^{-1} - 1\right)x^2\right| - \eta \tag{3.36}$$

Using both Equation 3.22 and Equation 3.36 in the MATLAB and Simulink simulation environment, results of the parameter estimation performance can be seen in Section 4.1.1.

3.3.2 First Order Nonlinear Model Implementation

Parameter identification derivation for the first order nonlinear system described in Section 2.1.2 is show, consider the system equation as defined in Equation 3.37 with uncertain parameter $\hat{a}$.

$$\begin{aligned} \dot{x} &= -\hat{a}\sin(y)x + Bu \\ \dot{y} &= -Cyx + \sin(\omega t) \end{aligned} \tag{3.37}$$

The candidate Lyapunov function is defined by equation 3.38

$$V = \tfrac{1}{2}s^2 \geq 0 \tag{3.38}$$

The Lyapunov function described in Equation 3.38 is positive semi-definite for all time, the sliding surface is then defined as Equation 3.57.

$$\begin{aligned} s = \tfrac{1}{2}\left(x^2 - x_m^2\right) + \lambda \int \tfrac{1}{2}\left(x^2 - x_m^2\right) \\ + \tfrac{1}{2}\left(y^2 - y_m^2\right) + \lambda \int \tfrac{1}{2}\left(y^2 - y_m^2\right) \end{aligned} \tag{3.39}$$

To ensure no movement off of the sliding surface set $\dot{s} = 0$, shown in Equation 3.40. An additional $\left[K\{(x\dot{x} - x_m\dot{x}_m) + (y\dot{y} - y_m\dot{y}_m)\} + \eta\right]\dot{\hat{a}}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined to provide Lyapunov stability and guarantee convergence. The η parameter in the sliding surface derivative is added to prevent a divide by zero condition in the numerical simulation.

$$\begin{aligned} \dot{s} = 0 = (x\dot{x} - x_m\dot{x}_m) + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) \\ + (y\dot{y} - y_m\dot{y}_m) + \lambda\tfrac{1}{2}\left(y^2 - y_m^2\right) \\ - \left[K\{(x\dot{x} - x_m\dot{x}_m) + (y\dot{y} - y_m\dot{y}_m)\} + \eta\right]\dot{\hat{a}} \end{aligned} \tag{3.40}$$

Substitute equation 3.37 into Equation 3.40 and re-arrange yields Equation 3.41

$$\begin{aligned} \dot{\hat{a}} = \tfrac{1}{K\{(x\dot{x} - x_m\dot{x}_m) + (y\dot{y} - y_m\dot{y}_m)\} + \eta}[-\hat{a}\sin(y)x^2 + Bux - \\ x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) + Cy^2x + y\sin(\omega t) - y_m\dot{y}_m + \lambda\tfrac{1}{2}\left(y^2 - y_m^2\right)] \end{aligned} \tag{3.41}$$

Equation 3.41 defines the equation that will be programmed into MATLAB and Simulink to determine $\hat{a}$. To satisfy the Lyapunov theorem of global stability as defined in Section 3.2.3, gains need to be chosen so that Equation 3.42 is true, ensuring stability and convergence.

$$\dot{V} = s\dot{s} < 0 \tag{3.42}$$

A more convenient choice still satisfying Equation 3.42 is Equation 3.43.

$$s\dot{s} \leq -\left(K + \eta\right)|s| \tag{3.43}$$

Where K and η are strictly positive values. Consider the actual system defined by Equation 3.44.

$$\begin{aligned} \dot{x} &= -a\sin(y)x + Bu \\ \dot{y} &= -Cyx + \sin(\omega t) \end{aligned} \tag{3.44}$$

Substituting Equation 3.44 into Equation 3.40:

$$\begin{aligned} \dot{s} = -a\sin(y)x^2 + Bux - x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) \\ + Cy^2x + y\sin(\omega t) - y_m\dot{y}_m + \lambda\tfrac{1}{2}\left(y^2 - y_m^2\right) \\ - \left[K\{(x\dot{x} - x_m\dot{x}_m) + (y\dot{y} - y_m\dot{y}_m)\} + \eta\right]\dot{\hat{a}} = 0 \end{aligned} \tag{3.45}$$

Now substituting Equation 3.41 into Equation 3.45:

$$\dot{s} = (\hat{a} - a)\sin(y)x^2 \tag{3.46}$$

Substituting Equation 3.46 into Equation 3.43:

$$K\,|s| \leq \left[(a - \hat{a})\sin(y)x^2\right] s - \eta\,|s| \tag{3.47}$$

The parameter a is unknown but assumed to be bounded and defined by the Equation 3.48.

$$a_{low} \leq a \leq a_{upp} \tag{3.48}$$

Assume the parameter a is the geometric mean of the bounds defined in Equation 3.48

$$a = \sqrt{a_{low}a_{upp}} \tag{3.49}$$

Applying Equation 3.49 to Equation 3.48 results in Equation 3.50.

$$\alpha \leq \frac{a}{\hat{a}} \leq \alpha^{-1} \tag{3.50}$$

Where:

$$\alpha = \sqrt{a_{upp}/a_{low}} \tag{3.51}$$

Equation 3.47 can be re-written as Equation 3.52.

$$K\,|s| \leq \left[\hat{a}\left(\frac{a}{\hat{a}} - 1\right)\sin(y)x^2\right] s - \eta\,|s| \tag{3.52}$$

Comparing the right side of the equality constraint of equation 3.50 which is taken to be the upper bound limit and the most conservative estimate for the unknown parameter to Equation 3.52 results in Equation 3.53.

$$K\,|s| \leq \left|\hat{a}\left(\alpha^{-1} - 1\right)\sin(y)x^2\right| |s| - \eta\,|s| \tag{3.53}$$

The value of K using the equality constraint of Equation 3.53 becomes Equation 3.54.

$$K = \left|\hat{a}\left(\alpha^{-1} - 1\right)\sin(y)x^2\right| - \eta \tag{3.54}$$

Using both Equation 3.41 and Equation 3.54 in the MATLAB and Simulink simulation environment, results of the parameter estimation performance can be seen in Section 4.1.2.

3.3.3 Second Order Model Implementation

Parameter identification derivation for the second order system described in Section 2.1.3 is shown, consider the system equation as defined in Equation 3.55 with uncertain parameters $\hat{a}_1$ and $\hat{a}_2$.

$$\ddot{x} = -\hat{a}_2\dot{x} - \hat{a}_1 x + u \tag{3.55}$$

The candidate Lyapunov function is defined by Equation 3.56

$$V = \tfrac{1}{2}s^2 \geq 0 \tag{3.56}$$

The Lyapunov function described in Equation 3.56 is positive semi-definite for all time, the sliding surfaces for both parameters $\hat{a}_1$ and $\hat{a}_2$ are then defined as Equation 3.57. Two sliding surfaces are built for the two parameters that are being estimated, sliding surface s_1 is for estimating parameter $\hat{a}_1$ and sliding surface s_2 is for estimating parameter $\hat{a}_2$.

$$\begin{aligned} s_1 &= \tfrac{1}{2}\left(\dot{x}^2 - \dot{x}_m^2\right) + \tfrac{1}{2}\lambda_1 \int \left(x^2 - x_m^2\right) \\ s_2 &= \tfrac{1}{2}\left(\dot{x}^2 - \dot{x}_m^2\right) + \tfrac{1}{2}\lambda_2 \int \left(x^2 - x_m^2\right) \end{aligned} \tag{3.57}$$

To ensure no movement off of the sliding surface set all $\dot{s} = 0$, shown in Equation 3.58. An additional $\left[K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined to provide Lyapunov stability and guarantee convergence. The η parameter in the sliding surface derivative is added to prevent a divide by zero condition in the numerical simulation.

$$\begin{aligned} \dot{s}_1 &= \left(\dot{x}\ddot{x} - \dot{x}_m\ddot{x}_m\right) + \lambda_1\left(x\dot{x} - x_m\dot{x}_m\right) - \left[k_1\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}}_1 = 0 \\ \dot{s}_2 &= \left(\dot{x}\ddot{x} - \dot{x}_m\ddot{x}_m\right) + \lambda_2\left(x\dot{x} - x_m\dot{x}_m\right) - \left[k_2\left(x\dot{x} - x_m\dot{x}_m\right) + \eta\right]\dot{\hat{a}}_2 = 0 \end{aligned} \tag{3.58}$$

Substitute Equation 3.55 into Equation 3.58 and re-arrange yields Equation 3.59

$$\begin{aligned} \dot{\hat{a}}_1 &= \tfrac{1}{k_1(x\dot{x} - x_m\dot{x}_m) + \eta}\left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_1\left(x\dot{x} - x_m\dot{x}_m\right)\right] \\ \dot{\hat{a}}_2 &= \tfrac{1}{k_2(x\dot{x} - x_m\dot{x}_m) + \eta}\left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_2\left(x\dot{x} - x_m\dot{x}_m\right)\right] \end{aligned} \tag{3.59}$$

Equation 3.59 defines the equation that will be programmed into MATLAB and Simulink to determine $\hat{a}_1$ and $\hat{a}_2$. To satisfy the Lyapunov theorem of global stability as defined in Section 3.2.3, gains need to be chosen so that Equation 3.60 is true, ensuring stability and convergence.

$$\dot{V} = s\dot{s} < 0 \tag{3.60}$$

A more convenient choice still satisfying Equation 3.60 is Equation 3.61.

$$s\dot{s} \leq -(K+\eta)|s| \tag{3.61}$$

Where K and η are strictly positive values. Consider the actual system defined by Equation 3.62.

$$\ddot{x} = -a_2\dot{x} - a_1 x + u \tag{3.62}$$

Substituting Equation 3.62 into Equation 3.58:

$$\begin{aligned}\dot{s}_1 &= -a_2\dot{x}^2 - a_1 x\dot{x} + u\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_1(x\dot{x} - x_m\dot{x}_m) - [k_1(x\dot{x} - x_m\dot{x}_m) + \eta]\,\dot{\hat{a}}_1 = 0 \\ \dot{s}_2 &= -a_2\dot{x}^2 - a_1 x\dot{x} + u\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_2(x\dot{x} - x_m\dot{x}_m) - [k_2(x\dot{x} - x_m\dot{x}_m) + \eta]\,\dot{\hat{a}}_2 = 0\end{aligned} \tag{3.63}$$

Now substituting Equation 3.59 into Equation 3.63:

$$\dot{s}_1 = \dot{s}_2 = (\hat{a}_1 - a_1)\,x\dot{x} + (\hat{a}_2 - a_2)\,\dot{x}^2 \tag{3.64}$$

Substituting Equation 3.64 into Equation 3.61:

$$K|s| \leq \left[(a_1 - \hat{a}_1)\,x\dot{x} + (a_2 - \hat{a}_2)\,\dot{x}^2\right] s - \eta|s| \tag{3.65}$$

The parameter a is unknown but assumed to be bounded and defined by the Equation 3.66.

$$a_{low} \leq a \leq a_{upp} \tag{3.66}$$

Assume the parameter a is the geometric mean of the bounds defined in Equation 3.66

$$a = \sqrt{a_{low}a_{upp}} \tag{3.67}$$

Applying Equation 3.67 to Equation 3.66 results in Equation 3.68.

$$\alpha \leq \frac{a}{\hat{a}} \leq \alpha^{-1} \tag{3.68}$$

Where:

$$\alpha = \sqrt{a_{upp}/a_{low}} \tag{3.69}$$

Equation 3.65 can be re-written as Equation 3.70.

$$K|s| \leq \left[\hat{a}_1\left(\frac{a_1}{\hat{a}_1} - 1\right)x\dot{x} + \hat{a}_2\left(\frac{a_2}{\hat{a}_2} - 1\right)\dot{x}^2\right] s - \eta|s| \tag{3.70}$$

Comparing the right side of the equality constraint of Equation 3.68 which is taken to be the upper bound limit and the most conservative estimate for the unknown parameter to Equation 3.70 results in Equation 3.71.

$$K\left|s\right| \leq \left|\hat{a}_1\left(\alpha_1^{-1}-1\right)x\dot{x}+\hat{a}_2\left(\alpha_2^{-1}-1\right)\dot{x}^2\right|\left|s\right|-\eta\left|s\right| \tag{3.71}$$

The value of K using the equality constraint of Equation 3.71 becomes Equation 3.72.

$$K=\left|\hat{a}_1\left(\alpha_1^{-1}-1\right)x\dot{x}+\hat{a}_2\left(\alpha_2^{-1}-1\right)\dot{x}^2\right|-\eta \tag{3.72}$$

Using both Equation 3.59 and Equation 3.72 in the MATLAB and Simulink simulation environment, results of the parameter estimation performance can be seen in Section 4.1.3.

3.3.4 Second Order Nonlinear Model Implementation

Parameter identification derivation for the second order nonlinear system described in Section 2.1.4 is show, consider the system equation is defined as Equation 3.73 with uncertain parameters $\hat{a}_1$ and $\hat{a}_2$.

$$\ddot{x}=-\hat{a}_2\dot{x}-\hat{a}_1x^2+u \tag{3.73}$$

The candidate Lyapunov function is defined by Equation 3.74

$$V=\tfrac{1}{2}s^2\geq 0 \tag{3.74}$$

The Lyapunov function described in Equation 3.74 is positive semi-definite for all time, the sliding surfaces for both parameters $\hat{a}_1$ and $\hat{a}_2$ are then defined as Equation 3.75. Two sliding surfaces are built for the two parameters that are being estimated, sliding surface s_1 is for estimating parameter $\hat{a}_1$ and sliding surface s_2 is for estimating parameter $\hat{a}_2$.

$$\begin{aligned} s_1&=\tfrac{1}{2}\left(\dot{x}^2-\dot{x}_m^2\right)+\tfrac{1}{2}\lambda_1\textstyle\int\left(x^2-x_m^2\right)\\ s_2&=\tfrac{1}{2}\left(\dot{x}^2-\dot{x}_m^2\right)+\tfrac{1}{2}\lambda_2\textstyle\int\left(x^2-x_m^2\right)\end{aligned} \tag{3.75}$$

To ensure no movement off of the sliding surface set all $\dot{s}=0$, shown in Equation 3.76. An additional $\left[K\left(x\dot{x}-x_m\dot{x}_m\right)+\eta\right]\dot{\hat{a}}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined to provide Lyapunov stability and guarantee

convergence. The η parameter in the sliding surface derivative is added to prevent a divide by zero condition in the numerical simulation.

$$\begin{aligned}\dot{s}_1 &= (\dot{x}\ddot{x} - \dot{x}_m\ddot{x}_m) + \lambda_1 (x\dot{x} - x_m\dot{x}_m) - [k_1 (x\dot{x} - x_m\dot{x}_m) + \eta]\,\dot{\hat{a}}_1 = 0 \\ \dot{s}_2 &= (\dot{x}\ddot{x} - \dot{x}_m\ddot{x}_m) + \lambda_2 (x\dot{x} - x_m\dot{x}_m) - [k_2 (x\dot{x} - x_m\dot{x}_m) + \eta]\,\dot{\hat{a}}_2 = 0\end{aligned} \tag{3.76}$$

Substitute Equation 3.73 into Equation 3.76 and re-arrange yields Equation 3.77

$$\begin{aligned}\dot{\hat{a}}_1 &= \tfrac{1}{k_1(x\dot{x} - x_m\dot{x}_m)+\eta} \left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x^2\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_1 (x\dot{x} - x_m\dot{x}_m)\right] \\ \dot{\hat{a}}_2 &= \tfrac{1}{k_2(x\dot{x} - x_m\dot{x}_m)+\eta} \left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x^2\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_2 (x\dot{x} - x_m\dot{x}_m)\right]\end{aligned} \tag{3.77}$$

Equation 3.77 defines the equation that will be programmed into MATLAB and Simulink to determine $\hat{a}_1$ and $\hat{a}_2$. To satisfy the Lyapunov theorem of global stability as defined in Section 3.2.3, gains need to be chosen so that Equation 3.78 is true, ensuring stability and convergence.

$$\dot{V} = s\dot{s} < 0 \tag{3.78}$$

A more convenient choice still satisfying Equation 3.78 is Equation 3.79.

$$s\dot{s} \leq -(K + \eta)\,|s| \tag{3.79}$$

Where K and η are strictly positive values. Consider the actual system defined by Equation 3.80.

$$\ddot{x} = -a_2\dot{x} - a_1 x^2 + u \tag{3.80}$$

Substituting Equation 3.80 into Equation 3.76:

$$\begin{aligned}\dot{s}_1 &= -a_2\dot{x}^2 - a_1 x^2\dot{x} + u\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_1 (x\dot{x} - x_m\dot{x}_m) - [k_1 (x\dot{x} - x_m\dot{x}_m) + \eta]\,\dot{\hat{a}}_1 = 0 \\ \dot{s}_2 &= -a_2\dot{x}^2 - a_1 x^2\dot{x} + u\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_2 (x\dot{x} - x_m\dot{x}_m) - [k_2 (x\dot{x} - x_m\dot{x}_m) + \eta]\,\dot{\hat{a}}_2 = 0\end{aligned} \tag{3.81}$$

Now substituting Equation 3.77 into Equation 3.81:

$$\dot{s}_1 = \dot{s}_2 = (\hat{a}_1 - a_1)\, x^2\dot{x} + (\hat{a}_2 - a_2)\,\dot{x}^2 \tag{3.82}$$

Substituting Equation 3.82 into Equation 3.79:

$$K\,|s| \leq \left[(a_1 - \hat{a}_1)\, x^2\dot{x} + (a_2 - \hat{a}_2)\,\dot{x}^2\right] s - \eta\,|s| \tag{3.83}$$

The parameter a is unknown but assumed to be bounded and defined by the Equation 3.84.

$$a_{low} \leq a \leq a_{upp} \tag{3.84}$$

Assume the parameter a is the geometric mean of the bounds defined in Equation 3.84

$$a = \sqrt{a_{low} a_{upp}} \tag{3.85}$$

Applying Equation 3.85 to Equation 3.84 results in Equation 3.86.

$$\alpha \leq \frac{a}{\hat{a}} \leq \alpha^{-1} \tag{3.86}$$

Where:

$$\alpha = \sqrt{a_{upp}/a_{low}} \tag{3.87}$$

Equation 3.83 can be re-written as Equation 3.88.

$$K\,|s| \leq \left[\hat{a}_1 \left(\frac{a_1}{\hat{a}_1} - 1\right) x^2 \dot{x} + \hat{a}_2 \left(\frac{a_2}{\hat{a}_2} - 1\right) \dot{x}^2\right] s - \eta\,|s| \tag{3.88}$$

Comparing the right side of the equality constraint of equation 3.86 which is taken to be the upper bound limit and the most conservative estimate for the unknown parameter to Equation 3.88 results in Equation 3.89.

$$K\,|s| \leq \left|\hat{a}_1 \left(\alpha_1^{-1} - 1\right) x^2 \dot{x} + \hat{a}_2 \left(\alpha_2^{-1} - 1\right) \dot{x}^2\right| |s| - \eta\,|s| \tag{3.89}$$

The value of K using the equality constraint of Equation 3.89 becomes Equation 3.90.

$$K = \left|\hat{a}_1 \left(\alpha_1^{-1} - 1\right) x^2 \dot{x} + \hat{a}_2 \left(\alpha_2^{-1} - 1\right) \dot{x}^2\right| - \eta \tag{3.90}$$

Using both Equation 3.77 and Equation 3.90 in the MATLAB and Simulink simulation environment, results of the parameter estimation performance can be seen in Section 4.1.4.

3.3.5 Summary of Equations

A summary of the final equations derived for the proof of concepts models is included in this section, the equations shown below are used in the MATLAB and Simulink analysis to prove the parameters can be properly identified, these equations are the result of the derivations included in the proof of concept Section 3.3. These equations allow for a system parameter to be estimated in a real-time environment and the analysis performed as part of this work can be found in the results Section 4.1.

First Order Model Equation Summary

The final equation for estimation of $\hat{a}$ parameter in the first order system, defined in Section 2.1.1, is shown as Equation 3.91 where the K value is defined as Equation 3.92 and the η term is a small number added to prevent a divide by zero condition in the simulation. The simulation results for these equations are shown in Section 4.1.1.

$$\dot{\hat{a}} = \frac{1}{K\left(x\dot{x} - x_m\dot{x}_m\right) + \eta}\left[ux - \hat{a}x^2 - x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right)\right] \tag{3.91}$$

Where:

$$K = \left|\hat{a}\left(\alpha^{-1} - 1\right)x^2\right| - \eta \tag{3.92}$$

The α parameter in Equation 3.92 is shown in Equation 3.93.

$$\alpha = \sqrt{a_{upp}/a_{low}} \tag{3.93}$$

First Order Nonlinear Model Equation Summary

The final equation for estimation of $\hat{a}$ in the first order nonlinear system, defined in Section 2.1.2, is shown as Equation 3.94 where the K value is defined as Equation 3.95 and the η term is a small number added to prevent a divide by zero condition in the simulation. The simulation results for these equations are shown in Section 4.1.2.

$$\begin{aligned}\dot{\hat{a}} = {} & \tfrac{1}{K\{(x\dot{x} - x_m\dot{x}_m) + (y\dot{y} - y_m\dot{y}_m)\} + \eta}[-\hat{a}\sin(y)x^2 + Bux - \\ & x_m\dot{x}_m + \lambda\tfrac{1}{2}\left(x^2 - x_m^2\right) + Cy^2x + y\sin(\omega t) - y_m\dot{y}_m + \lambda\tfrac{1}{2}\left(y^2 - y_m^2\right)]\end{aligned} \tag{3.94}$$

Where:

$$K = \left|\hat{a}\left(\alpha^{-1} - 1\right)\sin(y)x^2\right| - \eta \tag{3.95}$$

The α parameter in Equation 3.92 follows the same form as Equation 3.93.

Second Order Model Equation Summary

The final equation for estimation of $\hat{a}_1$ and $\hat{a}_2$ in the second order system, defined in Section 2.1.3, is shown as Equation 3.96 where the K value is defined as Equation 3.97

and the η term is a small number added to prevent a divide by zero condition in the simulation. The simulation results for these equations are shown in Section 4.1.3.

$$\begin{aligned}\dot{\hat{a}}_1 &= \frac{1}{k_1(x\dot{x}-x_m\dot{x}_m)+\eta}\left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_1\left(x\dot{x} - x_m\dot{x}_m\right)\right] \\ \dot{\hat{a}}_2 &= \frac{1}{k_2(x\dot{x}-x_m\dot{x}_m)+\eta}\left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_2\left(x\dot{x} - x_m\dot{x}_m\right)\right]\end{aligned} \tag{3.96}$$

$$K = \left|\hat{a}_1\left(\alpha_1^{-1} - 1\right)x\dot{x} + \hat{a}_2\left(\alpha_2^{-1} - 1\right)\dot{x}^2\right| - \eta \tag{3.97}$$

The α parameters in Equation 3.97 follows the same form as Equation 3.93.

Second Order Nonlinear Model Equation Summary

The final equation for estimation of $\hat{a}_1$ and $\hat{a}_2$ in the second order nonlinear system, defined in Section 2.1.4, is shown as Equation 3.98 where the K value is defined as Equation 3.99 and the η term is a small number added to prevent a divide by zero condition in the simulation. The simulation results for these equations are shown in Section 4.1.4.

$$\begin{aligned}\dot{\hat{a}}_1 &= \frac{1}{k_1(x\dot{x}-x_m\dot{x}_m)+\eta}\left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x^2\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_1\left(x\dot{x} - x_m\dot{x}_m\right)\right] \\ \dot{\hat{a}}_2 &= \frac{1}{k_2(x\dot{x}-x_m\dot{x}_m)+\eta}\left[\dot{x}u - \hat{a}_2\dot{x}^2 - \hat{a}_1 x^2\dot{x} - \dot{x}_m\ddot{x}_m + \lambda_2\left(x\dot{x} - x_m\dot{x}_m\right)\right]\end{aligned} \tag{3.98}$$

Where:

$$K = \left|\hat{a}_1\left(\alpha_1^{-1} - 1\right)x^2\dot{x} + \hat{a}_2\left(\alpha_2^{-1} - 1\right)\dot{x}^2\right| - \eta \tag{3.99}$$

The α parameters in Equation 3.99 follows the same form as Equation 3.93.

3.4 Aircraft Parameter Identification Development

The parameter identification methodology that has been developed in Section 3.2.2 will be used to estimate aerodynamic coefficients that describe the flight of an aircraft in real-time. During flight maneuvers, the aerodynamic parameters or coefficients are adjusted until the sliding surface and the derivative of the sliding surface are driven to zero. Once the sliding surface and its derivative have been driven to zero, the parameter estimation is complete and the best estimate is available. The results of the development in this section can be seen in the aircraft parameter estimation results Section 4.2.

3.4.1 Longitudinal Parameters

The longitudinal aerodynamic coefficients are the non-dimensional pitching moment (C_m), lift force (C_L), and drag force (C_D) coefficients. Equation 3.100 defines the longitudinal aerodynamic coefficients to be identified by the parameter estimator. The parameters with the subscript $()_T$ represent the best available coefficient value obtained from the F-16 nonlinear aerodynamic model. The remaining parameters are incremental updates to the nonlinear aerodynamic model that will be determined through the parameter identification process. The assumed general form of the model is shown in Equation 3.100, these parameters are a subset of the possible parameters to estimate and may be changed for other aircraft applications.

$$\begin{aligned} C_m &= C_{m_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{m_\alpha}\Delta\alpha + \tfrac{\bar{c}}{2V_T}\left(\Delta C_{m_q}\Delta q\right) \\ C_L &= C_{L_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{L_\alpha}\Delta\alpha \\ C_D &= C_{D_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{D_\alpha}\Delta\alpha \end{aligned} \tag{3.100}$$

Equation 3.100 uses the $\Delta\alpha$ and Δq notation to define the change in variable from the trim condition. For the longitudinal case, these variables are defined by Equation 3.101.

$$\begin{aligned} \Delta\alpha &= \alpha - \alpha_{trim} \\ \Delta q &= q - q_{trim} \end{aligned} \tag{3.101}$$

C_m PID Development

The two coefficients that will be estimated in the pitching moment coefficient equation (C_m, defined in Equation 3.100) are the change in moment coefficient due to angle of attack change (ΔC_{m_α}) and the change in moment coefficient due to a change in pitch rate change (ΔC_{m_q}). The estimator development for these two parameters is defined in this section. The aircraft moment equations that have been derived in Section 2.15 will be used for the system model in the pitching moment coefficient estimator development. The aircraft moment matrix equations have been re-written here as Equation 3.102 with the unknown parameter $\hat{C}_m$ defined in Equation 3.103.

$$\begin{bmatrix} \hat{\dot{p}} \\ \hat{\dot{q}} \\ \hat{\dot{r}} \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}^{-1} \times \begin{bmatrix} \bar{q}SbC_l + qr\left(I_{yy} - I_{zz}\right) + \left(q^2 - r^2\right) I_{yz} + pqI_{xz} - prI_{xy} \\ \bar{q}S\bar{c}\hat{C}_m + pr\left(I_{zz} - I_{xx}\right) + \left(r^2 - p^2\right) I_{xz} + qrI_{xy} - pqI_{yz} \\ \bar{q}SbC_n + pq\left(I_{xx} - I_{yy}\right) + \left(p^2 - q^2\right) I_{xy} + prI_{yz} - qrI_{xz} \end{bmatrix} \tag{3.102}$$

$$\hat{C}_m = C_{m_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta\hat{C}_{m_\alpha}\Delta\alpha + \tfrac{\bar{c}}{2V_T}\left(\Delta\hat{C}_{m_q}\Delta q\right) \tag{3.103}$$

The candidate Lyapunov function is defined by Equation 3.104, this function is positive semi-definite for all time.

$$V = \tfrac{1}{2}s^2 \geq 0 \tag{3.104}$$

The sliding surface for the unknown parameter, $\hat{C}_{m_\alpha}$, is defined as the following Equation 3.105.

$$\begin{aligned} s = {} & \tfrac{1}{2}\left(p^2 - p_m^2\right) + \lambda \int \tfrac{1}{2}\left(p^2 - p_m^2\right) + \\ & \tfrac{1}{2}\left(q^2 - q_m^2\right) + \lambda \int \tfrac{1}{2}\left(q^2 - q_m^2\right) + \\ & \tfrac{1}{2}\left(r^2 - r_m^2\right) + \lambda \int \tfrac{1}{2}\left(r^2 - r_m^2\right) \end{aligned} \tag{3.105}$$

To ensure there is no movement off of the sliding surface set $\dot{s} = 0$, shown in Equation 3.106. An additional $\left[K\{(p\dot{p} - p_m\dot{p}_m) + (q\dot{q} - q_m\dot{q}_m) + (r\dot{r} - r_m\dot{r}_m)\} + \eta\right]\dot{\hat{C}}_{m_\alpha}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined to provide

Lyapunov stability and guarantee convergence. The η parameter in the sliding surface derivative is added to prevent a divide by zero condition in the numerical simulation.

$$\begin{aligned}\dot{s} = 0 = &(p\dot{p} - p_m\dot{p}_m) + \tfrac{\lambda}{2}(p^2 - p_m^2) + (q\dot{q} - q_m\dot{q}_m) + \\ &\tfrac{\lambda}{2}(q^2 - q_m^2) + (r\dot{r} - r_m\dot{r}_m) + \tfrac{\lambda}{2}(r^2 - r_m^2) - \\ &[K\{(p\dot{p} - p_m\dot{p}_m) + (q\dot{q} - q_m\dot{q}_m) + (r\dot{r} - r_m\dot{r}_m)\} + \eta]\,\dot{\hat{C}}_{m_\alpha}\end{aligned} \tag{3.106}$$

Now substituting Equation 3.102 into Equation 3.106 and re-arranging yields Equation 3.107.

$$\dot{\hat{C}}_{m_\alpha} = \frac{num\_\dot{\hat{C}}_{m_\alpha}}{K\left[(p\dot{p} - p_m\dot{p}_m) + (q\dot{q} - q_m\dot{q}_m) + (r\dot{r} - r_m\dot{r}_m)\right] + \eta} \tag{3.107}$$

Where:

$$num\_\dot{\hat{C}}_{m_\alpha} = \begin{bmatrix} \left(p\dot{\hat{p}} - p_m\dot{p}_m\right) + \tfrac{\lambda}{2}(p^2 - p_m^2) + \left(q\dot{\hat{q}} - q_m\dot{q}_m\right) \\ +\tfrac{\lambda}{2}(q^2 - q_m^2) + \left(r\dot{\hat{r}} - r_m\dot{r}_m\right) + \tfrac{\lambda}{2}(r^2 - r_m^2) \end{bmatrix} \tag{3.108}$$

Equation 3.107 defines the equation that will be programmed into MATLAB and Simulink to determine the aerodynamic coefficient $\hat{C}_{m_\alpha}$. To satisfy the Lyapunov theorem of global stability as defined in Section 3.2.3, gains need to be chosen so that Equation 3.109 is true, ensuring stability and convergence.

$$\dot{V} = s\dot{s} < 0 \tag{3.109}$$

A more convenient choice still satisfying Equation 3.109 is Equation 3.110.

$$s\dot{s} \leq -(K + \eta)\,|s| \tag{3.110}$$

Where K and η are strictly positive values. Consider the actual system defined by Equation 3.111.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}^{-1} \begin{bmatrix} \bar{q}SbC_l + qr(I_{yy} - I_{zz}) + (q^2 - r^2)I_{yz} + pqI_{xz} - prI_{xy} \\ \bar{q}S\bar{c}C_m + pr(I_{zz} - I_{xx}) + (r^2 - p^2)I_{xz} + qrI_{xy} - pqI_{yz} \\ \bar{q}SbC_n + pq(I_{xx} - I_{yy}) + (p^2 - q^2)I_{xy} + prI_{yz} - qrI_{xz} \end{bmatrix} \tag{3.111}$$

Substituting Equation 3.111 into Equation 3.106:

$$\begin{aligned}\dot{s} = 0 = p\dot{p} - p_m\dot{p}_m + \tfrac{\lambda}{2}\left(p^2 - p_m^2\right) + q\dot{q} - q_m\dot{q}_m + \\ \tfrac{\lambda}{2}\left(q^2 - q_m^2\right) + r\dot{r} - r_m\dot{r}_m + \tfrac{\lambda}{2}\left(r^2 - r_m^2\right) - \\ \left[K\{(p\dot{p} - p_m\dot{p}_m) + (q\dot{q} - q_m\dot{q}_m) + (r\dot{r} - r_m\dot{r}_m)\} + \eta\right]\dot{\hat{C}}_{m_\alpha}\end{aligned} \tag{3.112}$$

Now substituting Equation 3.107 into Equation 3.112:

$$\dot{s} = 0 = p\left(\dot{p} - \dot{\hat{p}}\right) + q\left(\dot{q} - \dot{\hat{q}}\right) + r\left(\dot{r} - \dot{\hat{r}}\right) \tag{3.113}$$

The following steps (Equation 3.114 through Equation 3.118) will reduce the complexity of Equation 3.113, the first step will be to re-define the inverse of the inertia tensor matrix into a generic form. Equation 3.114 is the inverse of the inertia tensor matrix that was originally presented in the aircraft moment Equation 3.102

$$\begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}^{-1} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{3.114}$$

Substituting the newly defined matrix (Equation 3.114) into the aircraft system Equation 3.111 yields Equation 3.115.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} \bar{q}SbC_l + qr\left(I_{yy} - I_{zz}\right) + \left(q^2 - r^2\right)I_{yz} + pqI_{xz} - prI_{xy} \\ \bar{q}S\bar{c}C_m + pr\left(I_{zz} - I_{xx}\right) + \left(r^2 - p^2\right)I_{xz} + qrI_{xy} - pqI_{yz} \\ \bar{q}SbC_n + pq\left(I_{xx} - I_{yy}\right) + \left(p^2 - q^2\right)I_{xy} + prI_{yz} - qrI_{xz} \end{bmatrix} \tag{3.115}$$

Now that Equation 3.115 has been defined, the subtraction of the states inside of the parenthesis in Equation 3.113 will be reduced to a much simpler equation, this is defined in Equation 3.116.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} - \begin{bmatrix} \dot{\hat{p}} \\ \dot{\hat{q}} \\ \dot{\hat{r}} \end{bmatrix} = \begin{bmatrix} a_{12}\left(\bar{q}S\bar{c}C_m - \bar{q}S\bar{c}\hat{C}_m\right) \\ a_{22}\left(\bar{q}S\bar{c}C_m - \bar{q}S\bar{c}\hat{C}_m\right) \\ a_{32}\left(\bar{q}S\bar{c}C_m - \bar{q}S\bar{c}\hat{C}_m\right) \end{bmatrix} = \begin{bmatrix} C_p\left(C_m - \hat{C}_m\right) \\ C_q\left(C_m - \hat{C}_m\right) \\ C_r\left(C_m - \hat{C}_m\right) \end{bmatrix} \tag{3.116}$$

Where:

$$\begin{bmatrix} C_p \\ C_q \\ C_r \end{bmatrix} = \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} \bar{q} S \bar{c} \tag{3.117}$$

Now substituting Equation 3.116 back into the sliding surface derivative (Equation 3.113) results in the following:

$$\dot{s} = 0 = (pC_p + qC_q + rC_r)\left(C_m - \hat{C}_m\right) \tag{3.118}$$

The $C_m - \hat{C}_m$ is defined for all of the unknown parameters in Equation 3.119.

$$\left(C_m - \hat{C}_m\right) = \left(\Delta C_{m_\alpha} - \Delta \hat{C}_{m_\alpha}\right)\Delta\alpha + \left(\Delta C_{m_q} - \Delta \hat{C}_{m_q}\right)\frac{\bar{c}}{2V_T}\Delta q \tag{3.119}$$

Now that the complexity reduction in $\dot{s}$ has been completed, substituting Equation 3.118 into Equation 3.110 and re-arranging:

$$K\,|s| \leq \left[(pC_p + qC_q + rC_r)\left(\hat{C}_m - C_m\right)\right] s - \eta\,|s| \tag{3.120}$$

The η value added in Equation 3.120 is a very small number that will be used for the simulation to prevent a divide by zero condition. The parameter manipulation equations shown below will be applied to both of the unknown parameters, ($\Delta\hat{C}_{m_\alpha}$ and $\Delta\hat{C}_{m_q}$). The parameter C_m is unknown but assumed to be bounded and defined by the Equation 3.121.

$$C_{mlow} \leq C_m \leq C_{mupp} \tag{3.121}$$

Assume the parameter C_m is the geometric mean of the bounds defined in equation 3.121

$$C_m = \sqrt{C_{mlow} C_{mupp}} \tag{3.122}$$

Applying Equation 3.122 to Equation 3.121 results in the Equation 3.123.

$$\mu_{Cm} \leq \frac{C_m}{\hat{C}_m} \leq \mu_{C_m}^{-1} \tag{3.123}$$

Where:

$$\mu_{C_m} = \sqrt{C_{mupp} / C_{mlow}} \tag{3.124}$$

Equation 3.120 can be re-written as Equation 3.125.

$$K\,|s| \leq \left[\left(pC_p + qC_q + rC_r \right) \hat{C}_m \left(1 - \tfrac{C_m}{\hat{C}_m} \right) \right] s - \eta\,|s| \tag{3.125}$$

Comparing the right side of the inequality constraint of equation 3.123, which is taken to be the upper bound limit and the most conservative estimate for the unknown parameter, to Equation 3.125 results in Equation 3.126.

$$K\,|s| \leq \left| \left(pC_p + qC_q + rC_r \right) \hat{C}_m \left(1 - \mu_{C_m}^{-1} \right) \right| |s| - \eta\,|s| \tag{3.126}$$

The value of K using the equality constraint of Equation 3.126 becomes Equation 3.127.

$$K = \left| \left(pC_p + qC_q + rC_r \right) \hat{C}_m \left(1 - \mu_{C_m}^{-1} \right) \right| - \eta \tag{3.127}$$

Expanding Equation 3.127 to include both of the unknown parameters, ($\Delta\hat{C}_{m_\alpha}$ and $\Delta\hat{C}_{m_q}$) is shown in Equation 3.128.

$$K = \left| \left(pC_p + qC_q + rC_r \right) \begin{pmatrix} \hat{C}_{m_\alpha} \left(1 - \mu_{C_{m_\alpha}}^{-1} \right) \Delta\alpha + \\ \hat{C}_{m_q} \left(1 - \mu_{C_{m_q}}^{-1} \right) \frac{\bar{c}}{2V_T} \Delta q \end{pmatrix} \right| - \eta \tag{3.128}$$

The analysis completed in Section 3.4.1 for $\Delta\hat{C}_{m_\alpha}$ can also be applied to the $\Delta\hat{C}_{m_q}$ parameter in the C_m Equation 3.103. The derivation for the $\Delta\hat{C}_{m_q}$ parameter follows the same steps as shown above in Equation 3.102 through Equation 3.128, a summary of PID derivations can be found in the aircraft PID equation summary, Section 3.4.2.

C_L PID Development

The coefficient that will be estimated in the lifting moment coefficient equation (C_L, defined in Equation 3.100) is the change in lift coefficient due to angle of attack change (ΔC_{L_α}). The estimator derivation for these two parameters will be defined in this section.

The aircraft kinematic equations that have been derived in Section 2.22 will be used for the system model in the lifting moment coefficient estimator development. The aircraft kinematic equations have been re-written here in Equation 3.129.

$$\begin{aligned}
&\dot{\alpha} = q - (p\cos\alpha + r\sin\alpha)\tan\beta - \tfrac{LOM}{V_T\cos\beta} + \\
&\tfrac{g}{V_T\cos\beta}\left(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha\right) \\
&\dot{\beta} = p\sin\alpha - r\cos\alpha + \tfrac{1}{V_T}\left(YOM\cos\beta + DOM\sin\beta\right) + \\
&\tfrac{g}{V_T}\left(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha\right) \\
&\dot{V}_T = YOM\sin\beta - DOM\cos\beta + \\
&g\left[\left(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha\right)\cos\beta + \cos\theta\sin\phi\sin\beta\right]
\end{aligned} \tag{3.129}$$

Where:

$$\begin{aligned}
&DOM = \tfrac{D - T\cos\alpha}{m}; \quad YOM = \tfrac{Y}{m}; \quad LOM = \tfrac{L + T\sin\alpha}{m} \\
&D = \bar{q}SC_D; \qquad Y = \bar{q}SC_Y; \qquad L = \bar{q}S\hat{C}_L
\end{aligned} \tag{3.130}$$

The unknown parameter $\hat{C}_L$ that will be estimated is defined in equation 3.131.

$$C_L = C_{L_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{L_\alpha}\Delta\alpha \tag{3.131}$$

The candidate Lyapunov function is defined by Equation 3.132, this function is positive semi-definite for all time.

$$V = \tfrac{1}{2}s^2 \geq 0 \tag{3.132}$$

The sliding surface for the C_{L_α} is defined as the following Equation 3.133.

$$\begin{aligned}
&s = \tfrac{1}{2}\left(\alpha^2 - \alpha_m^2\right) + \lambda\int\tfrac{1}{2}\left(\alpha^2 - \alpha_m^2\right) + \\
&\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) + \lambda\int\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right)dr + \\
&\tfrac{1}{2}\left(\beta^2 - \beta_m^2\right) + \lambda\int\tfrac{1}{2}\left(\beta^2 - \beta_m^2\right)
\end{aligned} \tag{3.133}$$

To ensure there is no movement off of the sliding surface set $\dot{s} = 0$, shown in Equation 3.134. An additional $\left[K\{(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\} + \eta\right]\dot{\hat{C}}_{L_\alpha}$ term has been added to the sliding surface derivative, this additional term is included so that an equation for $\dot{\hat{a}}$ can be defined. The K value in this equation will be determined

to provide Lyapunov stability and guarantee convergence. The η parameter in the sliding surface derivative is added to prevent a divide by zero condition in the numerical simulation.

$$\begin{aligned}&\dot{s} = 0 = (\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \tfrac{\lambda}{2}(\alpha^2 - \alpha_m^2) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \\ &\tfrac{\lambda}{2}\left(V_T^2 - V_{T_m}^2\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right) + \tfrac{\lambda}{2}(\beta^2 - \beta_m^2) - \\ &\left[K\{(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\} + \eta\right]\dot{\hat{C}}_{L_\alpha}\end{aligned} \tag{3.134}$$

Now substituting Equation 3.129 into Equation 3.134 results in Equation 3.135.

$$\begin{aligned}\dot{s} = \alpha q - \alpha\tan\beta\,(p\cos\alpha + r\sin\alpha) - \tfrac{\alpha\bar{q}S\hat{C}_{L_\alpha} + T\sin\alpha}{mV_T\cos\beta} \\ + \tfrac{\alpha g}{V_T\cos\beta}(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) - \alpha_m\dot{\alpha}_m + \lambda\tfrac{1}{2}(\alpha^2 - \alpha_m^2) \\ + V_T YOM\sin\beta \\ - V_T DOM\cos\beta + V_T g\left[(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\cos\beta + \cos\theta\sin\phi\sin\beta\right] \\ - V_{T_m}\dot{V}_{T_m} + \lambda\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) \\ + \beta p\sin\alpha - \beta r\cos\alpha + \tfrac{\beta}{V_T}(YOM\cos\beta + DOM\sin\beta) \\ + \tfrac{\beta g}{V_T}(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha) \\ - \beta_m\dot{\beta}_m + \lambda\tfrac{1}{2}(\beta^2 - \beta_m^2) \\ - \left[K\{(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\} + \eta\right]\dot{\hat{C}}_{L_\alpha}\end{aligned} \tag{3.135}$$

Re-arranging and solving for $\dot{\hat{C}}_{L_\alpha}$ yields Equation 3.136.

$$\dot{\hat{C}}_{L_\alpha} = \frac{Num\_\dot{\hat{C}}_{L_\alpha}}{K\left[(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\right] + \eta} \tag{3.136}$$

Where $Num\_\dot{\hat{C}}_{L_\alpha}$ is defined in Equation 3.137.

$$\begin{aligned}
\dot{\hat{C}}_{L_\alpha} = Num_ = [&\alpha q - \alpha \tan\beta\,(p\cos\alpha + r\sin\alpha) - \tfrac{\alpha\bar{q}S\hat{C}_{L_\alpha} + T\sin\alpha}{mV_T\cos\beta} \\
&+ \tfrac{\alpha g}{V_T\cos\beta}\,(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) - \alpha_m\dot{\alpha}_m + \lambda\tfrac{1}{2}\,(\alpha^2 - \alpha_m^2) \\
&+ V_T YOM\sin\beta \\
&- V_T DOM\cos\beta + V_T g\,[(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\cos\beta + \cos\theta\sin\phi\sin\beta] \\
&- V_{T_m}\dot{V}_{T_m} + \lambda\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) \\
&+ \beta p\sin\alpha - \beta r\cos\alpha + \tfrac{\beta}{V_T}\,(YOM\cos\beta + DOM\sin\beta) \\
&+ \tfrac{\beta g}{V_T}\,(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha) \\
&- \beta_m\dot{\beta}_m + \lambda\tfrac{1}{2}\,(\beta^2 - \beta_m^2)]
\end{aligned} \tag{3.137}$$

Equation 3.136 defines the equation that will be programmed into the MATLAB and Simulink environments to determine the aerodynamic coefficient $\hat{C}_{L_\alpha}$. To satisfy the Lyapunov theorem of global stability as defined in Section 3.2.3, gains need to be chosen so that Equation 3.138 is true, ensuring stability, and convergence.

$$\dot{V} = s\dot{s} < 0 \tag{3.138}$$

A more convenient choice still satisfying Equation 3.138 is Equation 3.139.

$$s\dot{s} \leq -\,(K + \eta)\,|s| \tag{3.139}$$

Where K and η are strictly positive values. Consider the actual system defined by Equation 3.140.

$$\begin{aligned}
\dot{\alpha} &= q - (p\cos\alpha + r\sin\alpha)\tan\beta - \tfrac{LOM}{V_T\cos\beta} + \\
&\tfrac{g}{V_T\cos\beta}\,(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) \\
\dot{\beta} &= p\sin\alpha - r\cos\alpha + \tfrac{1}{V_T}\,(YOM\cos\beta + DOM\sin\beta) + \\
&\tfrac{g}{V_T}\,(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha) \\
\dot{V}_T &= YOM\sin\beta - DOM\cos\beta + \\
&g\,[(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\cos\beta + \cos\theta\sin\phi\sin\beta]
\end{aligned} \tag{3.140}$$

Where:

$$DOM = \frac{D - T\cos\alpha}{m}; \quad YOM = \frac{Y}{m}; \quad LOM = \frac{L + T\sin\alpha}{m}$$
$$D = \bar{q}SC_D; \qquad Y = \bar{q}SC_Y; \qquad L = \bar{q}SC_L \tag{3.141}$$

Substituting Equation 3.140 into Equation 3.134:

$$\begin{gathered}
\dot{s} = \alpha q - \alpha\tan\beta\left(p\cos\alpha + r\sin\alpha\right) - \frac{\alpha\bar{q}SC_{L_\alpha} + T\sin\alpha}{mV_T\cos\beta} \\
+\frac{\alpha g}{V_T\cos\beta}\left(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha\right) - \alpha_m\dot{\alpha}_m + \lambda\tfrac{1}{2}\left(\alpha^2 - \alpha_m^2\right) \\
+V_T YOM\sin\beta \\
-V_T DOM\cos\beta + V_T g\left[\left(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha\right)\cos\beta + \cos\theta\sin\phi\sin\beta\right] \\
-V_{T_m}\dot{V}_{T_m} + \lambda\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) \\
+\beta p\sin\alpha - \beta r\cos\alpha + \tfrac{\beta}{V_T}\left(YOM\cos\beta + DOM\sin\beta\right) \\
+\tfrac{\beta g}{V_T}\left(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha\right) \\
-\beta_m\dot{\beta}_m + \lambda\tfrac{1}{2}\left(\beta^2 - \beta_m^2\right) \\
-\left[K\{(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\} + \eta\right]\dot{\hat{C}}_{L_\alpha}
\end{gathered} \tag{3.142}$$

Now substituting Equation 3.136 into Equation 3.142:

$$\dot{s} = \frac{\alpha\bar{q}S\hat{C}_L - \alpha\bar{q}SC_L}{mV_T\cos\beta} \tag{3.143}$$

Substituting Equation 3.143 into Equation 3.139 and re-arranging:

$$K\left|s\right| \leq \left[\left(C_L - \hat{C}_L\right)z\right]s - \eta\left|s\right| \tag{3.144}$$

Where:

$$z = \frac{\alpha\bar{q}S}{mV_T\cos\beta} \tag{3.145}$$

The parameter C_L is unknown but assumed to be bounded and defined by Equation 3.146.

$$C_{L_{low}} \leq C_L \leq C_{L_{upp}} \tag{3.146}$$

Assume the parameter C_L is the geometric mean of the bounds defined in Equation 3.146

$$C_L = \sqrt{C_{L_{low}} C_{L_{upp}}} \tag{3.147}$$

Applying Equation 3.147 to Equation 3.146 results in Equation 3.148.

$$\mu \leq \frac{C_L}{\hat{C}_L} \leq \mu^{-1} \tag{3.148}$$

Where:

$$\mu = \sqrt{C_{L_{upp}} \big/ C_{L_{low}}} \tag{3.149}$$

Equation 3.144 can be re-written as Equation 3.150.

$$K\,|s| \leq \left[\hat{C}_L \left(\frac{C_L}{\hat{C}_L} - 1\right) x^2\right] s - \eta\,|s| \tag{3.150}$$

Comparing the right side of the inequality constraint of Equation 3.148, which is taken to be the upper bound limit and the most conservative estimate for the unknown parameter, to Equation 3.150 results in Equation 3.151.

$$K\,|s| \leq \left[\hat{C}_L \left(\mu^{-1} - 1\right) z\right] |s| - \eta\,|s| \tag{3.151}$$

The value of K using the equality constraint of Equation 3.151 becomes Equation 3.152.

$$K = \left[\hat{C}_L \left(\mu^{-1} - 1\right) z\right] - \eta \tag{3.152}$$

Expanding Equation 3.152 for C_{L_α} results in the new K value that is defined in Equation 3.153.

$$K = \left[\hat{C}_{L_\alpha} \left(\mu_{\hat{C}_{L_\alpha}}^{-1} - 1\right) z\right] - \eta \tag{3.153}$$

The η value added in Equation 3.152 is a very small number that will be used for the simulation to prevent a divide by zero condition. Both Equation 3.136 and Equation 3.153 will be used to calculate the unknown parameter in the C_L aerodynamic coefficient equation. The analysis completed in Section 3.4.1 for $\hat{C}_L$ can also be applied to $\hat{C}_{L_\alpha}$.

The derivation for the unknown parameters follows the same steps as shown above in Equation 3.129 through Equation 3.153, a summary of PID derivations can be found in the aircraft PID equation summary Section 3.4.2.

C_D PID Development

The only coefficient that will be estimated in the coefficient of drag equation (C_D, defined in Equation 3.100) is the change in drag coefficient due to angle of attack change (ΔC_{D_α}). The estimator derivation for this parameter is extremely similar to that of the lifting moment coefficient that was derived in Section 3.4.1. The entire derivation of ΔC_{D_α} will not be shown, however the highlights of the derivation will be shown in this section.

The aircraft kinematic equations that have been derived in Section 2.22 will be used for the system model in the coefficient of drag estimator development. The aircraft kinematic equations have been re-written here in Equation 3.154.

$$\begin{aligned}
&\dot{\alpha} = q - (p\cos\alpha + r\sin\alpha)\tan\beta - \tfrac{LOM}{V_T\cos\beta} + \\
&\tfrac{g}{V_T\cos\beta}\left(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha\right) \\
&\dot{\beta} = p\sin\alpha - r\cos\alpha + \tfrac{1}{V_T}\left(YOM\cos\beta + DOM\sin\beta\right) + \\
&\tfrac{g}{V_T}\left(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha\right) \\
&\dot{V}_T = YOM\sin\beta - DOM\cos\beta + \\
&g\left[\left(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha\right)\cos\beta + \cos\theta\sin\phi\sin\beta\right]
\end{aligned} \tag{3.154}$$

Where:

$$\begin{aligned}
&DOM = \tfrac{D - T\cos\alpha}{m}; \quad YOM = \tfrac{Y}{m}; \quad LOM = \tfrac{L + T\sin\alpha}{m} \\
&D = \bar{q}S\hat{C}_D; \qquad Y = \bar{q}SC_Y; \qquad L = \bar{q}SC_L
\end{aligned} \tag{3.155}$$

The unknown parameter $\hat{C}_D$ that will be estimated is defined in equation 3.156.

$$C_D = C_{D_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{D_\alpha}\Delta\alpha \tag{3.156}$$

The same Lyapunov function will be used as in the C_L parameter estimator development and the steps taken from Equation 3.132 to Equation 3.136 are repeated for the C_{D_α} estimator and will result in Equation 3.157.

$$\dot{\hat{C}}_{D_\alpha} = \frac{Num\_\dot{\hat{C}}_{D_\alpha}}{K\left[(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\right] + \eta} \tag{3.157}$$

Where $Num\_\dot{\hat{C}}_{D_\alpha}$ is shown in Equation 3.158.

$$\begin{gathered}
Num_\dot{\hat{C}}_{D_\alpha} = [\alpha q - \alpha\tan\beta\,(p\cos\alpha + r\sin\alpha) - \tfrac{\alpha LOM}{V_T\cos\beta} \\
+\tfrac{\alpha g}{V_T\cos\beta}\,(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha) - \alpha_m\dot{\alpha}_m + \lambda\tfrac{1}{2}\,(\alpha^2 - \alpha_m^2) \\
+V_T YOM\sin\beta \\
-V_T\left(\tfrac{\bar{q}S\hat{C}_D - T\cos\alpha}{m}\right)\cos\beta + V_T g\,[(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha)\cos\beta + \cos\theta\sin\phi\sin\beta] \\
-V_{T_m}\dot{V}_{T_m} + \lambda\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) \\
+\beta p\sin\alpha - \beta r\cos\alpha + \tfrac{\beta}{V_T}\left(YOM\cos\beta + \left(\tfrac{\bar{q}S\hat{C}_D - T\cos\alpha}{m}\right)\sin\beta\right) \\
+\tfrac{\beta g}{V_T}\,(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha) \\
-\beta_m\dot{\beta}_m + \lambda\tfrac{1}{2}\,(\beta^2 - \beta_m^2)]
\end{gathered} \tag{3.158}$$

Following the steps shown in Equation 3.138 through Equation 3.153 for the K derivation , we can repeat those same steps to come up with the K equation for the C_{D_α} estimator. After following these steps, the results is shown in Equation 3.159.

$$K = \left[\hat{C}_{D_\alpha}\left(\mu_{\hat{C}_{D_\alpha}}^{-1} - 1\right)z\right] - \eta \tag{3.159}$$

Where $\mu_{\hat{C}_{D_\alpha}}$ is defined by Equation 3.160 and z is defined by Equation 3.161.

$$\mu_{\hat{C}_{D_\alpha}} = \sqrt{C_{D_\alpha upp}/C_{D_\alpha low}} \tag{3.160}$$

$$z = \tfrac{\bar{q}S}{m}\left(V_T cos\beta - \tfrac{\beta\sin\beta}{V_T}\right) \tag{3.161}$$

3.4.2 Equation Summary

A summary of the final equations derived for the aircraft model is included in this section, the equations shown below are used in the MATLAB and Simulink analysis to prove the parameters can be properly identified, these equations are the result of the derivations included in the Longitudinal parameter derivation, Section 4.2.1. These equations allow for an aircraft parameter to be estimated in a real-time environment and the analysis performed as part of this work can be found in the results Section 4.2.

C_m Equation Summary

Showing Equation 3.103 here as Equation 3.162 illustrates the three parameters to be estimated as part of the C_m equation.

$$\hat{C}_m = C_{m_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta\hat{C}_{m_\alpha}\Delta\alpha + \tfrac{\bar{c}}{2V_T}\left(\Delta\hat{C}_{m_q}\Delta q\right) \tag{3.162}$$

$\Delta\hat{C}_{m_\alpha}$ Final Equation

The final equation for the estimation of $\Delta\hat{C}_{m_\alpha}$ is shown as Equation 3.163 where the K value is defined by Equation 3.165. The simulation results for this estimator can be found in Section 4.2.1.

$$\dot{\hat{C}}_{m_\alpha} = \frac{num\_\dot{\hat{C}}_{m_\alpha}}{K\left[(p\dot{p} - p_m\dot{p}_m) + (q\dot{q} - q_m\dot{q}_m) + (r\dot{r} - r_m\dot{r}_m)\right] + \eta} \tag{3.163}$$

Where:

$$num\_\dot{\hat{C}}_{m_\alpha} = \begin{bmatrix} \left(p\hat{\dot{p}} - p_m\dot{p}_m\right) + \frac{\lambda}{2}\left(p^2 - p_m^2\right) + \left(q\hat{\dot{q}} - q_m\dot{q}_m\right) \\ +\frac{\lambda}{2}\left(q^2 - q_m^2\right) + \left(r\hat{\dot{r}} - r_m\dot{r}_m\right) + \frac{\lambda}{2}\left(r^2 - r_m^2\right) \end{bmatrix} \tag{3.164}$$

And:

$$K = \left| (pC_p + qC_q + rC_r) \begin{pmatrix} \hat{C}_{m_\alpha}\left(1 - \mu_{C_{m_\alpha}}^{-1}\right)\Delta\alpha + \\ \hat{C}_{m_q}\left(1 - \mu_{C_{m_q}}^{-1}\right)\frac{\bar{c}}{2V_T}\Delta q \end{pmatrix} \right| - \eta \tag{3.165}$$

$\Delta\hat{C}_{m_q}$ Final Equation

The final equation for the estimation of $\Delta\hat{C}_{m_q}$ is shown as Equation 3.166 where the K value is defined by Equation 3.165. The simulation results for this estimator can be found in Section 4.2.1.

$$\dot{\hat{C}}_{m_q} = \frac{num\_\dot{\hat{C}}_{m_q}}{K\left[(p\dot{p} - p_m\dot{p}_m) + (q\dot{q} - q_m\dot{q}_m) + (r\dot{r} - r_m\dot{r}_m)\right] + \eta} \tag{3.166}$$

Where:

$$num\_\dot{\hat{C}}_{m_q} = \begin{bmatrix} \left(p\hat{\dot{p}} - p_m\dot{p}_m\right) + \frac{\lambda}{2}\left(p^2 - p_m^2\right) + \left(q\hat{\dot{q}} - q_m\dot{q}_m\right) \\ +\frac{\lambda}{2}\left(q^2 - q_m^2\right) + \left(r\hat{\dot{r}} - r_m\dot{r}_m\right) + \frac{\lambda}{2}\left(r^2 - r_m^2\right) \end{bmatrix} \tag{3.167}$$

C_L Equation Summary

Showing Equation 3.131 here as Equation 3.168 illustrates the two parameters to be estimated as part of the C_L equation.

$$C_L = C_{L_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{L_\alpha}\Delta\alpha \tag{3.168}$$

$\Delta\hat{C}_{L_\alpha}$ Final Equation

The final equation for the estimation of $\Delta\hat{C}_{L_\alpha}$ is shown as Equation 3.169. The simulation results for this estimator can be found in Section 4.2.1.

$$\dot{\hat{C}}_{L_\alpha} = \frac{Num\_\dot{\hat{C}}_{L_\alpha}}{K\left[(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\right] + \eta} \tag{3.169}$$

Where $Num\_\dot{\hat{C}}_{L_\alpha}$ is defined in Equation 3.170.

$$
\begin{aligned}
\dot{\hat{C}}_{L_\alpha} = Num\_\dot{\hat{C}}_{L_\alpha} = [&\alpha q - \alpha \tan\beta \left(p\cos\alpha + r\sin\alpha\right) - \tfrac{\alpha\bar{q}S\hat{C}_{L_\alpha} + T\sin\alpha}{mV_T\cos\beta} \\
&+ \tfrac{\alpha g}{V_T\cos\beta}\left(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha\right) - \alpha_m\dot{\alpha}_m + \lambda\tfrac{1}{2}\left(\alpha^2 - \alpha_m^2\right) \\
&+ V_T YOM\sin\beta \\
&- V_T DOM\cos\beta + V_T g\left[\left(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha\right)\cos\beta + \cos\theta\sin\phi\sin\beta\right] \\
&- V_{T_m}\dot{V}_{T_m} + \lambda\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) \\
&+ \beta p\sin\alpha - \beta r\cos\alpha + \tfrac{\beta}{V_T}\left(YOM\cos\beta + DOM\sin\beta\right) \\
&+ \tfrac{\beta g}{V_T}\left(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha\right) \\
&- \beta_m\dot{\beta}_m + \lambda\tfrac{1}{2}\left(\beta^2 - \beta_m^2\right)]
\end{aligned}
\tag{3.170}
$$

And where K is shown in Equation 3.171.

$$
K = \left[\hat{C}_{L_\alpha}\left(\mu_{\hat{C}_{L_\alpha}}^{-1} - 1\right)z\right] - \eta \tag{3.171}
$$

C_D Equation Summary

Showing Equation 3.156 here as Equation 3.172 illustrates the parameter to be estimated as part of the C_D equation.

$$
C_D = C_{D_T}(\alpha, h, m, q, \dot{\alpha}, \delta_e, \ldots) + \Delta C_{D_\alpha}\Delta\alpha \tag{3.172}
$$

$\Delta\hat{C}_{D_\alpha}$ Final Equation

The final equation for the estimation of $\Delta\hat{C}_{D_\alpha}$ is shown as Equation 3.173 where the K value is defined by Equation 3.175. The simulation results for this estimator can be found in Section 4.2.1.

$$
\dot{\hat{C}}_{D_\alpha} = \frac{Num\_\dot{\hat{C}}_{D_\alpha}}{K\left[\left(\alpha\dot{\alpha} - \alpha_m\dot{\alpha}_m\right) + \left(V_T\dot{V}_T - V_{T_m}\dot{V}_{T_m}\right) + \left(\beta\dot{\beta} - \beta_m\dot{\beta}_m\right)\right] + \eta} \tag{3.173}
$$

Where $Num\_\dot{\hat{C}}_{D_\alpha}$ is shown in Equation 3.174.

$$\begin{gathered}
\dot{\hat{N}}um_\hat{C}_{D_\alpha} = [\alpha q - \alpha \tan\beta \left(p\cos\alpha + r\sin\alpha\right) - \tfrac{\alpha LOM}{V_T\cos\beta} \\
+\tfrac{\alpha g}{V_T\cos\beta}\left(\cos\theta\cos\phi\cos\alpha + \sin\theta\sin\alpha\right) - \alpha_m\dot{\alpha}_m + \lambda\tfrac{1}{2}\left(\alpha^2 - \alpha_m^2\right) \\
+V_T YOM\sin\beta \\
-V_T\left(\tfrac{\bar{q}S\hat{C}_D - T\cos\alpha}{m}\right)\cos\beta + V_T g\left[\left(\cos\theta\cos\phi\sin\alpha - \sin\theta\cos\alpha\right)\cos\beta + \cos\theta\sin\phi\sin\beta\right] \\
-V_{T_m}\dot{V}_{T_m} + \lambda\tfrac{1}{2}\left(V_T^2 - V_{T_m}^2\right) \\
+\beta p\sin\alpha - \beta r\cos\alpha + \tfrac{\beta}{V_T}\left(YOM\cos\beta + \left(\tfrac{\bar{q}S\hat{C}_D - T\cos\alpha}{m}\right)\sin\beta\right) \\
+\tfrac{\beta g}{V_T}\left(\cos\theta\sin\phi\cos\beta + \sin\theta\sin\beta\cos\alpha - \cos\theta\cos\phi\sin\beta\sin\alpha\right) \\
-\beta_m\dot{\beta}_m + \lambda\tfrac{1}{2}\left(\beta^2 - \beta_m^2\right)]
\end{gathered} \tag{3.174}$$

And:

$$K = \left[\hat{C}_{D_\alpha}\left(\mu_{\hat{C}_{D_\alpha}}^{-1} - 1\right)z\right] - \eta \tag{3.175}$$

Chapter 4

Results

MATLAB and Simulink programs have been developed and simulated for each of the system models defined in Section 2. The parameter estimation routines for both the proof of concept systems, defined in Section 3.3, and the aircraft model, defined in Section 3.4, have been simulated to verify the proper operation of the estimation technique. This section includes the simulation results for each of the systems and provides parameters used and resulting plots. There are three main components to each of the simulations, the first is the reference system model, the second part is the estimator, and the final part of the simulation is the estimated system model with the newly estimated parameters. A simulation overview is shown in Figure 3.2 of Section 3.2.2.

4.1 Proof of Concept Results

4.1.1 First Order Model Results

The first order model defined in Section 2.1.1 along with the first order estimator derived in Section 3.3.1 have been simulated together to ensure proper convergence of the estimated $\hat{a}$ to the actual a parameter. This results section contains plots illustrating the convergence of the estimated parameter along with the tracking of both the model reference state X and the estimated state X_{est}, for various different inputs to the system and model parameters.

Table 4.1 shows five different test conditions and results that are documented in this section, the variables that can be changed are the actual value of a, the upper and lower

Table 4.1: First Order Model Test Case Table

Test Case No	Command	a	a_{low}	a_{upp}	λ	η	Converge Time
1	Step, mag 1	3	.5	7	20	.0001	.5 sec
2	Step, mag 1	3	.5	7	50	.0001	.4 sec
3	Step, mag 1	5	1	8	50	.0001	1.4 sec
4	Step, mag 3	5	1	8	50	.0001	1 sec
5	Step, mag 3	-1	-2	-.9	50	.0001	.1 sec

bounds of a, λ, and η. Modifying these parameters results in changes to the convergence characteristics of the estimator, the differences can be seen in the plots below.

Test Case 1

Figure 4.1 shows two plots, the plot on the left is the tracking of the actual system state X to the estimated system state X_{est} and the plot on the right shows a_{est} converging upon the actual value of a which in this case is 3. The plots show that the estimator successfully tracks the desired parameter and takes approximately a half of a second to converge upon the actual parameter being estimated. The resulting estimated state, tracks the actual state perfectly after the estimator has converged and the parameter has been properly estimated.

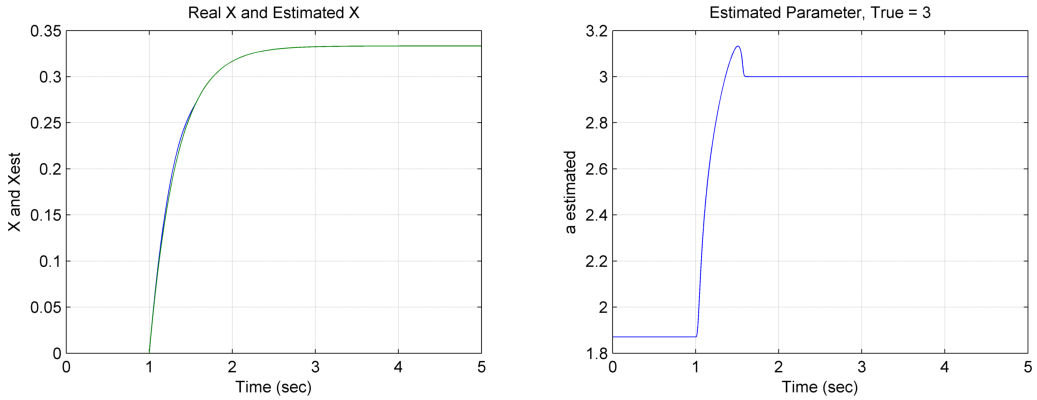

Figure 4.1: First Order Results Case 1, State Tracking and Parameter Convergence

Test Case 2

Figure 4.2 shows two plots, the plot on the left is the tracking of the actual system state X to the estimated system state X_{est} and the plot on the right shows a_{est} converging

upon the actual value of a which in this case is 3. The plots show that the estimator successfully tracks the desired parameter and takes less than half of a second to converge upon the actual parameter being estimated. The resulting estimated state, tracks the actual state perfectly after the estimator has converged and the parameter has been properly estimated. This case has a larger value of λ than case 1 and therefore is seen to converge slightly faster than the previous case, less than a half a second convergence time.

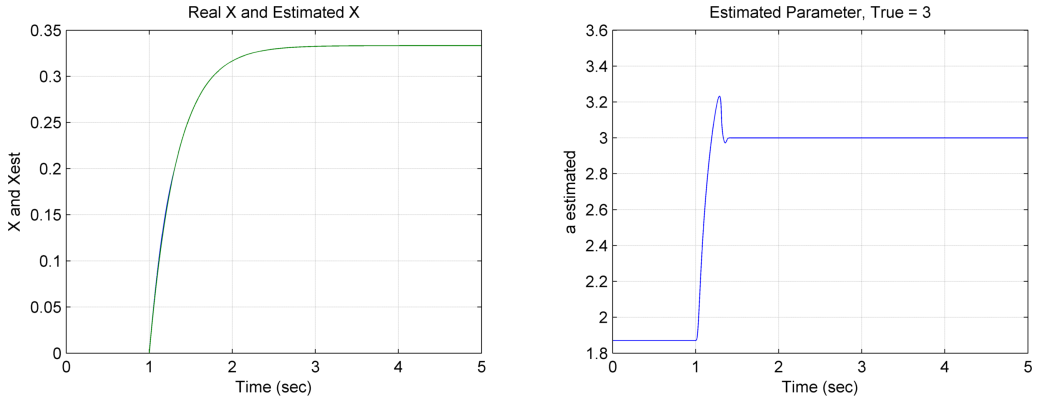

Figure 4.2: First Order Results Case 2, State Tracking and Parameter Convergence

Test Case 3

Figure 4.3 shows two plots, the plot on the left is the tracking of the actual system state X to the estimated system state X_{est} and the plot on the right shows a_{est} converging upon the actual value of a which in this case is 5. The plots show that the estimator successfully tracks the desired parameter and takes approximatively one and a half seconds to converge upon the actual parameter being estimated. The resulting estimated state, tracks the actual state perfectly after the estimator has converged and the parameter has been properly estimated. In this test case the actual a has changed to 5 and the time that is takes to converge upon that value increases significantly. This increase in convergence time is a result of the input step command to the system being a relatively low value of one. If the step command magnitude is increased, then the resulting convergence time will also increase. This is shown to be true in test case number four of the first order model.

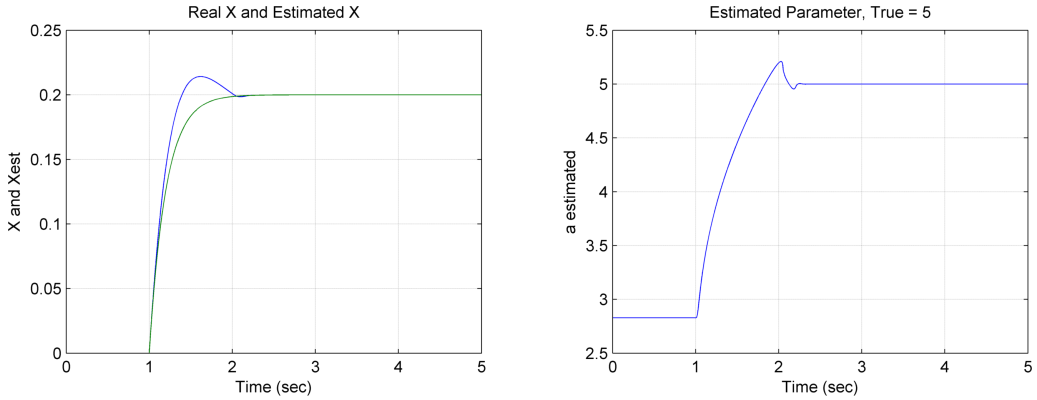

Figure 4.3: First Order Results Case 3, State Tracking and Parameter Convergence

Test Case 4

Figure 4.4 shows two plots, the plot on the left is the tracking of the actual system state X to the estimated system state X_{est} and the plot on the right shows a_{est} converging upon the actual value of a which in this case is 5. The plots show that the estimator successfully tracks the desired parameter and takes approximatively one second to converge upon the actual parameter being estimated. The resulting estimated state, tracks the actual state perfectly after the estimator has converged and the parameter has been properly estimated. The input was changed in this case to a magnitude of three, this allowed for a larger system excitation and a faster convergence to the actual value of a.

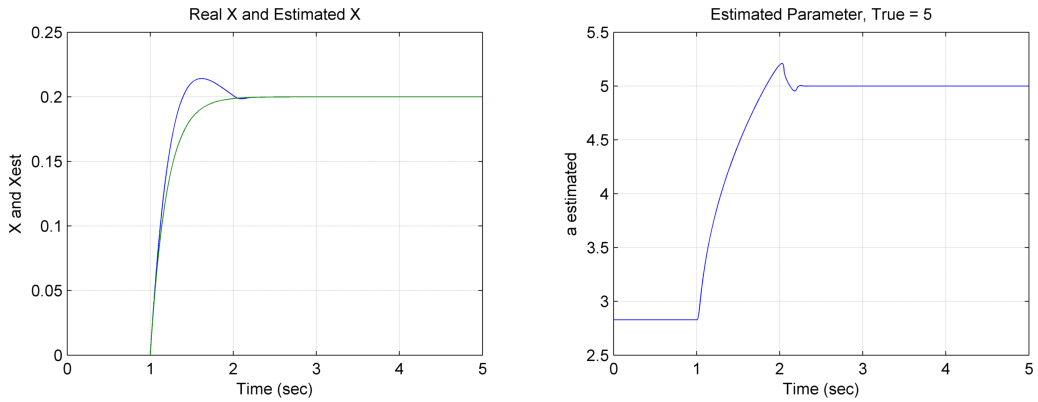

Figure 4.4: First Order Results Case 4, State Tracking and Parameter Convergence

Test Case 5

Figure 4.5 shows two plots, the plot on the left is the tracking of the actual system state X to the estimated system state X_{est} and the plot on the right shows a_{est} converging upon the actual value of a which in this case is -1. The plots show that the estimator successfully tracks the desired parameter and takes approximately a tenth of a second to converge upon the actual parameter being estimated. The resulting estimated state, tracks the actual state perfectly after the estimator has converged and the parameter has been properly estimated. This test case illustrated the ability of the parameter estimation routine to properly identify a negative unknown parameter, this is an important distinction from the traditional model reference adaptive estimation techniques. The initial bounds of the unknown parameter were set very close together, this allows for rapid convergence and a much faster estimation of the actual parameter value.

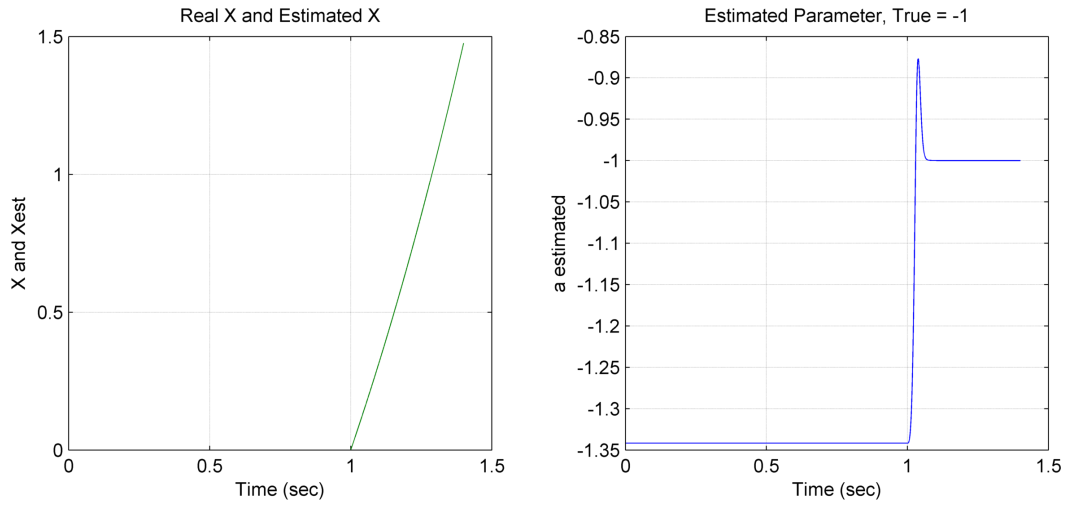

Figure 4.5: First Order Results Case 5, State Tracking and Parameter Convergence

4.1.2 First Order Nonlinear Model Results

The first order nonlinear model defined in Section 2.1.2 along with the first order nonlinear estimator derived in Section 3.3.2 have been simulated together to ensure proper convergence of the estimated $\hat{a}$ to the actual a parameter. This results section contains plots illustrating the convergence of the estimated parameter along with the tracking of both the model reference states X and Y with the estimated states X_{est} and Y_{est}, for various different inputs to the system and model parameters. The constants for the nonlinear model are defined as follows, $C = 4$, $B = 1.4$, and $\omega = .7rad/sec$.

Table 4.2 shows the four different test conditions and results that are documented in this section, the variables that can be changed are the input type, the actual value of a, the upper and lower bounds of a, λ, and η. These modification change the convergence characteristics of the estimator, the differences are illustrated in the plots below.

Table 4.2: First Order Nonlinear Model Test Case Table

Test Case No	Command	a	a_{low}	a_{upp}	λ	η	Converge Time
1	Sin, Amp=3, Fr=.7rad/sec	7	4	40	20	.001	2 sec
2	Sin, Amp=3, Fr=.5rad/sec	7	.1	40	20	.001	2.5 sec
3	Sin, Amp=7, Fr=.5rad/sec	7	.1	40	20	.001	1.4 sec
4	Sin, Amp=3, Fr=.7rad/sec	-3	-6	-1	20	.001	1 sec

Test Case 1

Figure 4.6 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the center is the tracking of the actual system state Y to the estimated system state Y_{est}, and the plot on the bottom shows a_{est} converging upon the actual value of a which in this case is 7. The plots show that the estimator successfully tracks the system states and takes approximately two seconds to converge upon the actual parameter being estimated. Both of the resulting estimated states, track the actual states perfectly after the estimator has converged and the parameter has been properly estimated. This convergence time for the first order nonlinear model is slightly larger than that of the first order model, this is due to the complex nature of the system model and the low amplitude input. Test case three (4.1.2) in this section will illustrate a faster convergence time with a more aggressive system input.

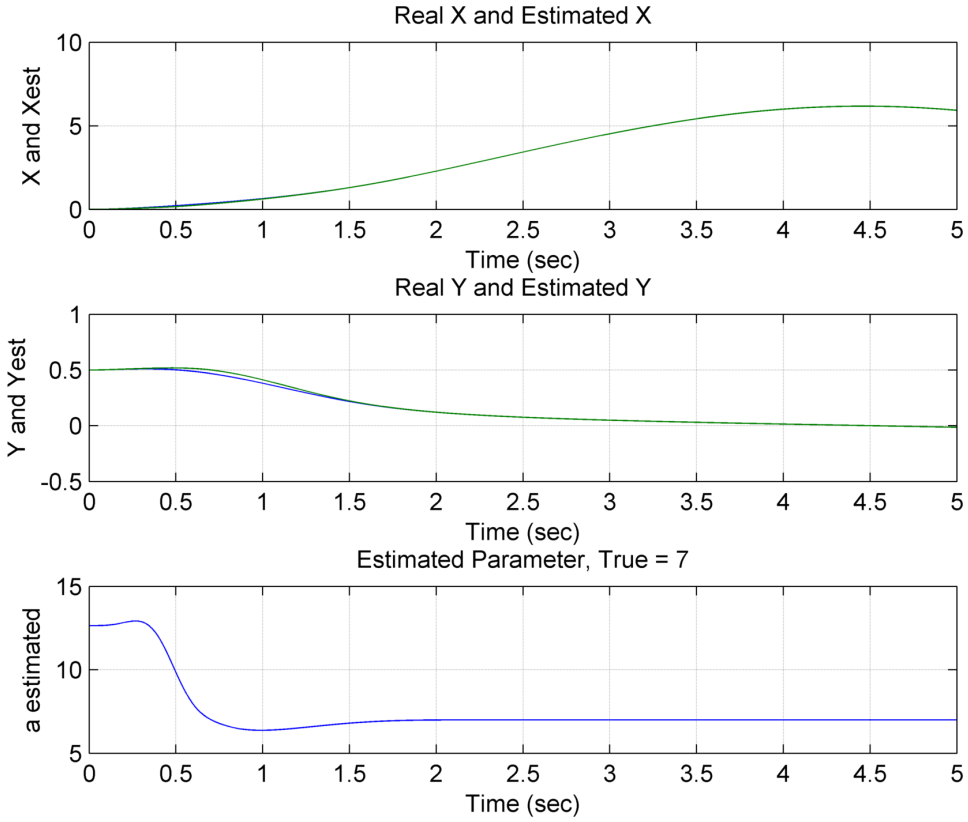

Figure 4.6: First Order Nonlinear Results Case 1, State Tracking and Parameter Convergence

Test Case 2

Figure 4.7 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the center is the tracking of the actual system state Y to the estimated system state Y_{est}, and the plot on the bottom shows a_{est} converging upon the actual value of a which in this case is 7. The plots show that the estimator successfully tracks the system states and takes approximately two and a half seconds to converge upon the actual parameter being estimated. Both of the resulting estimated states, track the actual states perfectly after the estimator has converged and the parameter has been properly estimated. The less aggressive system command results in a slower convergence time for the estimator, the command was lowered from a $.7rad/sec$ to a $.5rad/sec$ command with an increase in .5 seconds in convergence time.

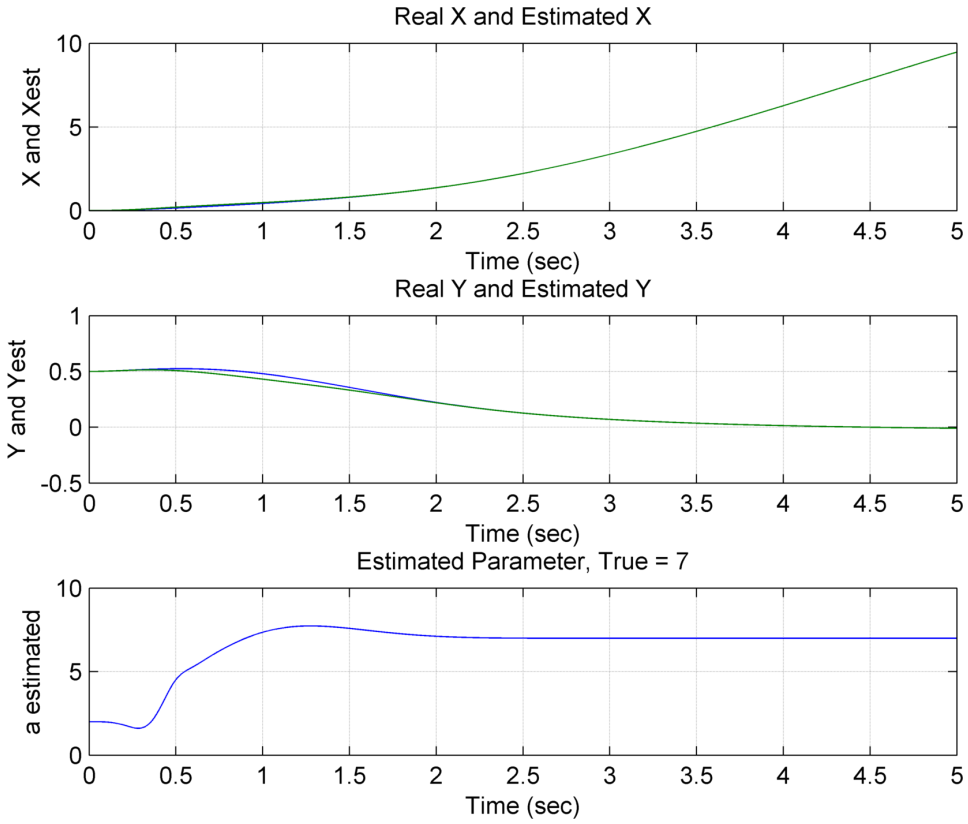

Figure 4.7: First Order Nonlinear Results Case 2, State Tracking and Parameter Convergence

Test Case 3

Figure 4.8 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the center is the tracking of the actual system state Y to the estimated system state Y_{est}, and the plot on the bottom shows a_{est} converging upon the actual value of a which in this case is 7. The plots show that the estimator successfully tracks the system states and takes less than one and a half seconds to converge upon the actual parameter being estimated. Both of the resulting estimated states, track the actual states perfectly after the estimator has converged and the parameter has been properly estimated. The system command inputs amplitude was increased for this case, which resulted in a significant drop in time to parameter convergence.

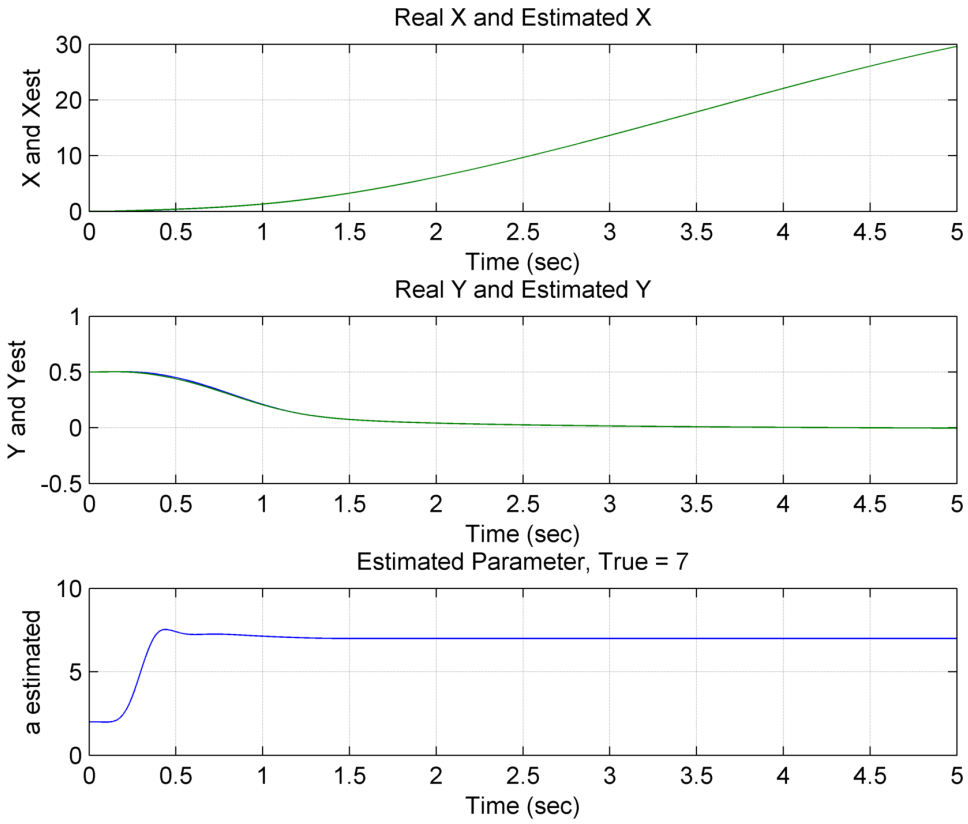

Figure 4.8: First Order Nonlinear Results Case 3, State Tracking and Parameter Convergence

Test Case 4

Figure 4.9 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the center is the tracking of the actual system state Y to the estimated system state Y_{est}, and the plot on the bottom shows a_{est} converging upon the actual value of a which in this case is -3. The plots show that the estimator successfully tracks the system states and takes approximately one second to converge upon the actual parameter being estimated. Both of the resulting estimated states track the actual states perfectly after the estimator has converged and the parameter has been properly estimated. This convergence time for the first order non-linear model is slightly larger than that of the first order model, this is due to the complex nature of the system model and the low amplitude input. This test case illustrated the ability of the parameter estimation routine to properly identify a negative unknown parameter for the first order nonlinear system, this is an important distinction from the traditional model reference adaptive estimation techniques.

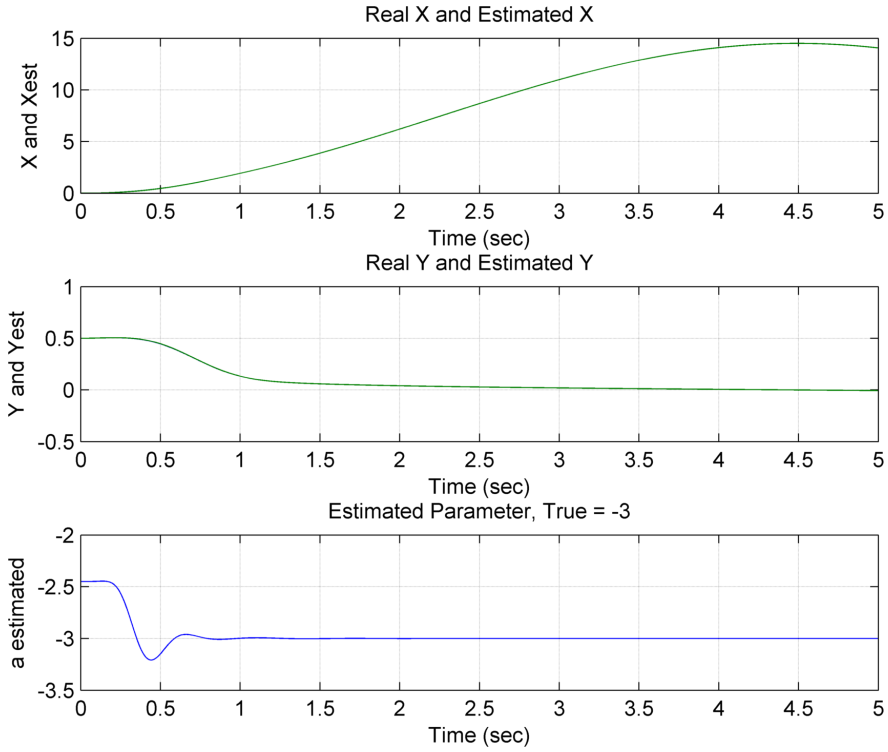

Figure 4.9: First Order Nonlinear Results Case 4, State Tracking and Parameter Convergence

4.1.3 Second Order Model Results

The second order model defined in Section 2.1.3 along with the second order estimator derived in Section 3.3.3 have been simulated together to ensure proper convergence of the estimated $\hat{a}_1$ to the actual a_1 and the estimated $\hat{a}_2$ to the actual a_2. This results section contains plots illustrating the convergence of the estimated parameters along with the tracking of the model reference state X with the estimated state X_{est}, for various different inputs to the system and model parameters.

Table 4.3 shows the five different test conditions and results that are documented in this section, the variables that can be changed are the input type, the actual value of a_1 and a_2, the initial a_1 and a_2,and λ for all second order model tests, η will be .001. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots below.

Table 4.3: Second Order Model Test Case Table

Test Case No	Command	a_1	a_2	a_{1est}	a_{2est}	λ	a_1Conv	a_2Conv
1	Step, Mag=6	8	6	4	2	100	.5 sec	.6 sec
2	Step, Mag=6	6	12	10	16	5	1.6 sec	1.5 sec
3	Sin, Amp=10, Fr=.2rad/sec	6	12	11	17	200	.7 sec	.9 sec
4	Sin, Amp=-10, Fr=.5rad/sec	18	6	12	0	100	.7 sec	.9 sec
5	Step, Mag=6	-9	-7	-5	-3	100	.5 sec	.6 sec

Test Case 1

Figure 4.10 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 8, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 6. The plots show that the estimator successfully tracks the system state and takes approximately half of a second to converge upon the actual parameter a_1 and it takes a little over half of a second to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly

estimated. This test case illustrates the estimators ability to correctly identify multiple parameters in a second order system with a system input of a step command.

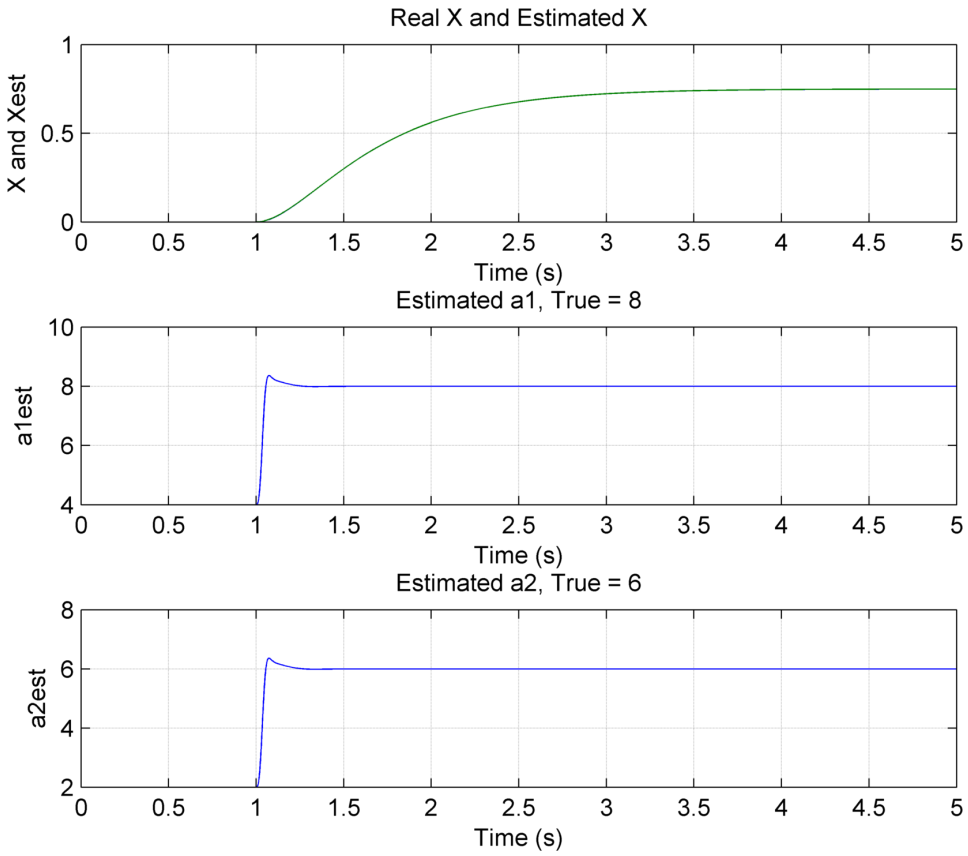

Figure 4.10: Second Order Model Results Case 1, State Tracking and Parameter Convergence

Test Case 2

Figure 4.11 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 6, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 12. The plots show that the estimator successfully tracks the system state and takes slightly over one and a half seconds to converge upon the actual parameter a_1 and it takes approximately one and a half seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly estimated. In this test case the λ parameter was changed to a very low

value, resulting in an expected increase in convergence time. This case also illustrates the ability of the estimator to successfully converge upon the actual parameters when the starting initial a_{1est} and a_{2est} are much higher than the actual values.

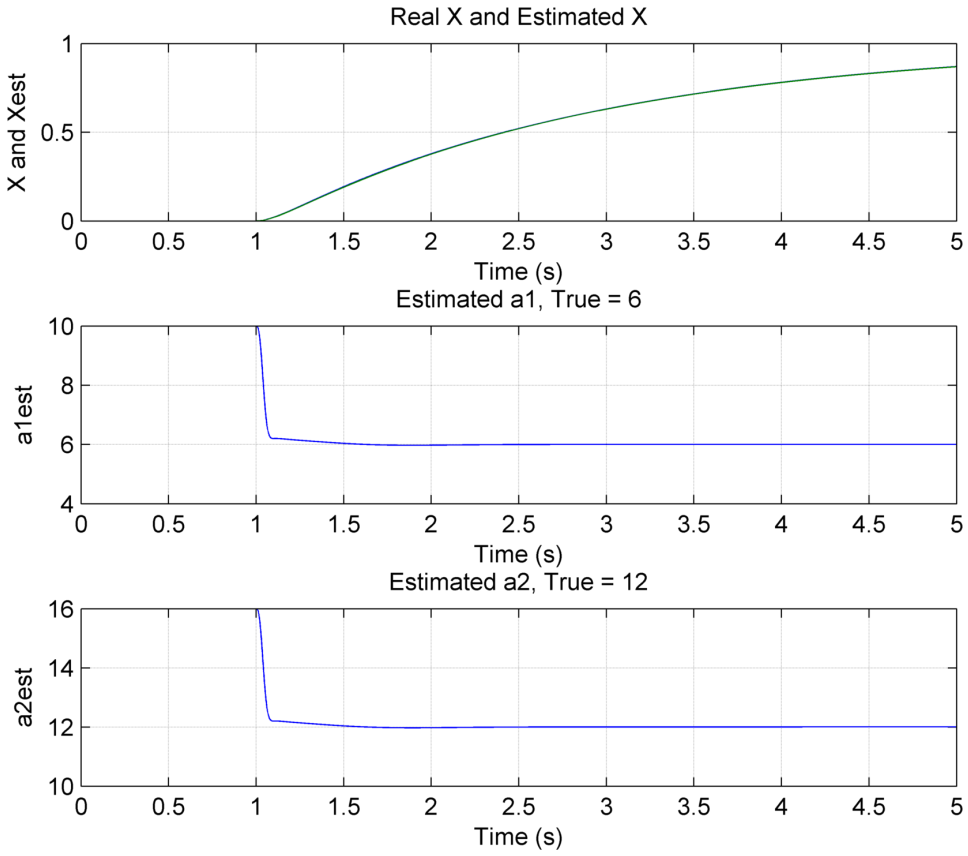

Figure 4.11: Second Order Model Results Case 2, State Tracking and Parameter Convergence

Test Case 3

Figure 4.12 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 6, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 12. The plots show that the estimator successfully tracks the system state and takes less than one second to converge upon the actual parameter a_1 and it takes approximately one second to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly estimated. The

input in this case is changed to a sin command input, this along with an increase in the λ parameter result in the fastest convergence time. This scenario excites this system very rapidly which leads to fast convergence and also shows the estimator will work with a system input of a sin command.

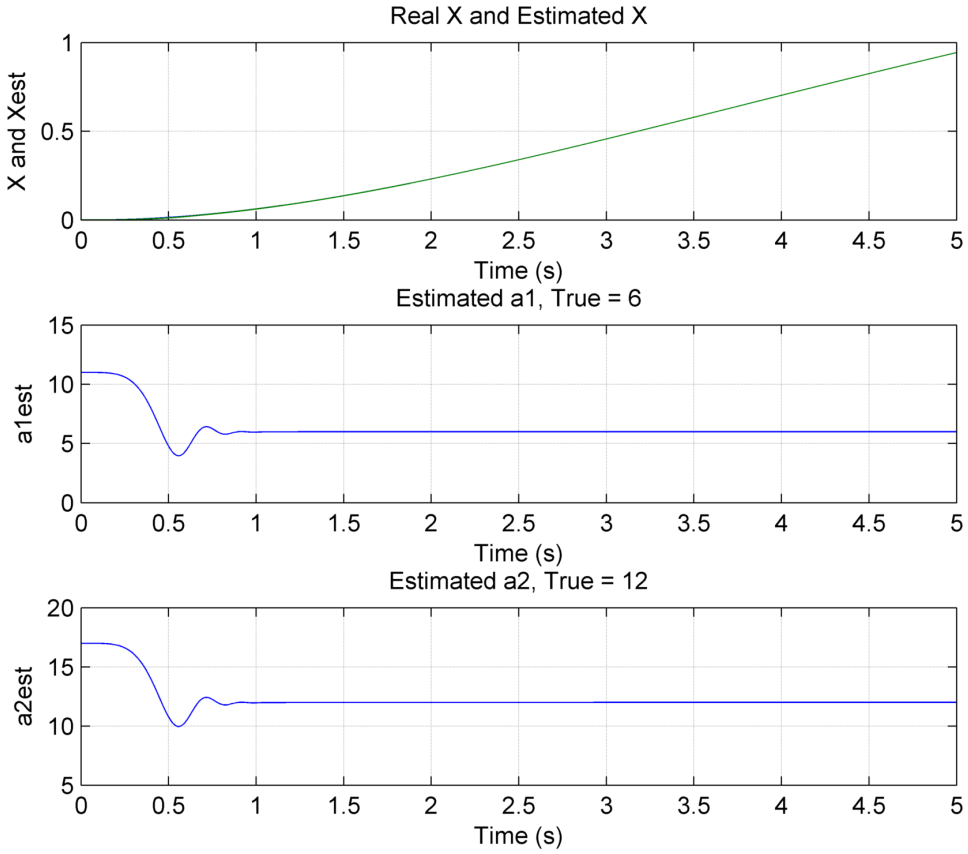

Figure 4.12: Second Order Model Results Case 3, State Tracking and Parameter Convergence

Test Case 4

Figure 4.13 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 18, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 6. The plots show that the estimator successfully tracks the system state and takes half a second to converge upon the actual parameter a_1 and it takes just over half a second to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the

estimator has converged and the two parameters have been properly estimated. This scenario is set up to provide a system command input that is negative, the input is a sin wave which will start in the negative direction. This negative direction and higher frequency command to the system result in a successful parameter identification for both a_1 and a_2.

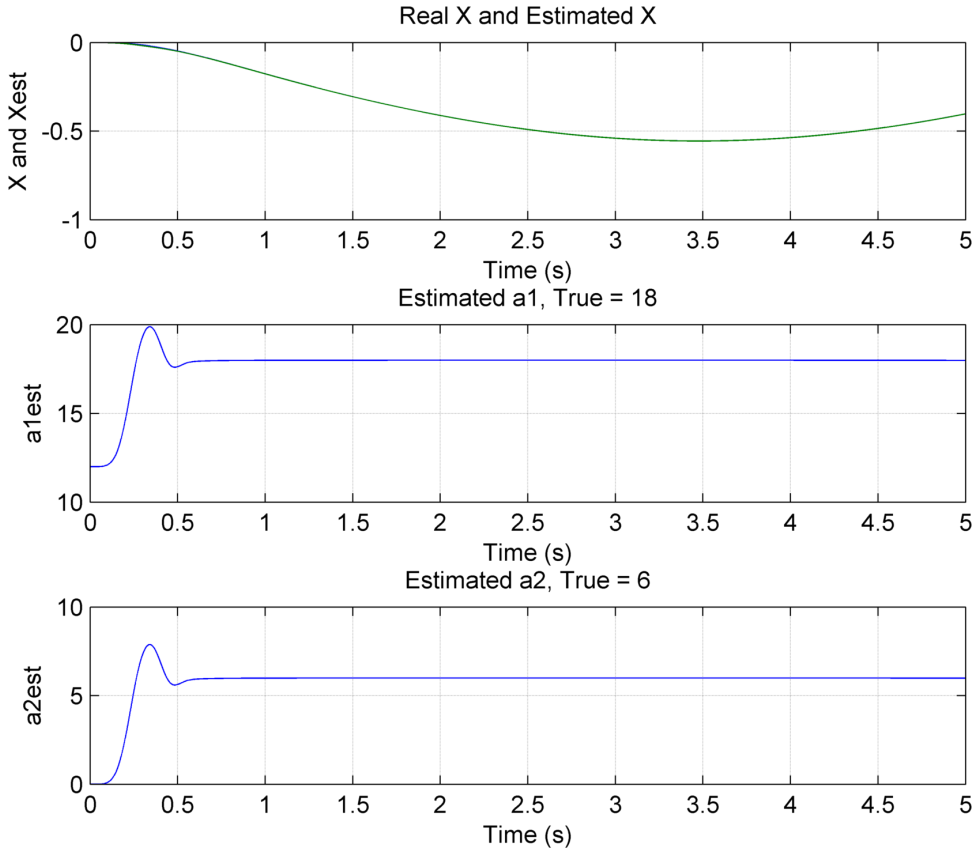

Figure 4.13: Second Order Model Results Case 4, State Tracking and Parameter Convergence

Test Case 5

Figure 4.14 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is -9, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is -7. The plots show that the estimator successfully tracks the system state and takes approximately half of a second to converge upon the actual parameter a_1 and it takes a little over half of

a second to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly estimated. This test case illustrates the estimators ability to correctly identify multiple parameters in a second order system with a system input of a step command in addition to the ability of the parameter estimation routine to properly identify a negative unknown parameter for a second order system.

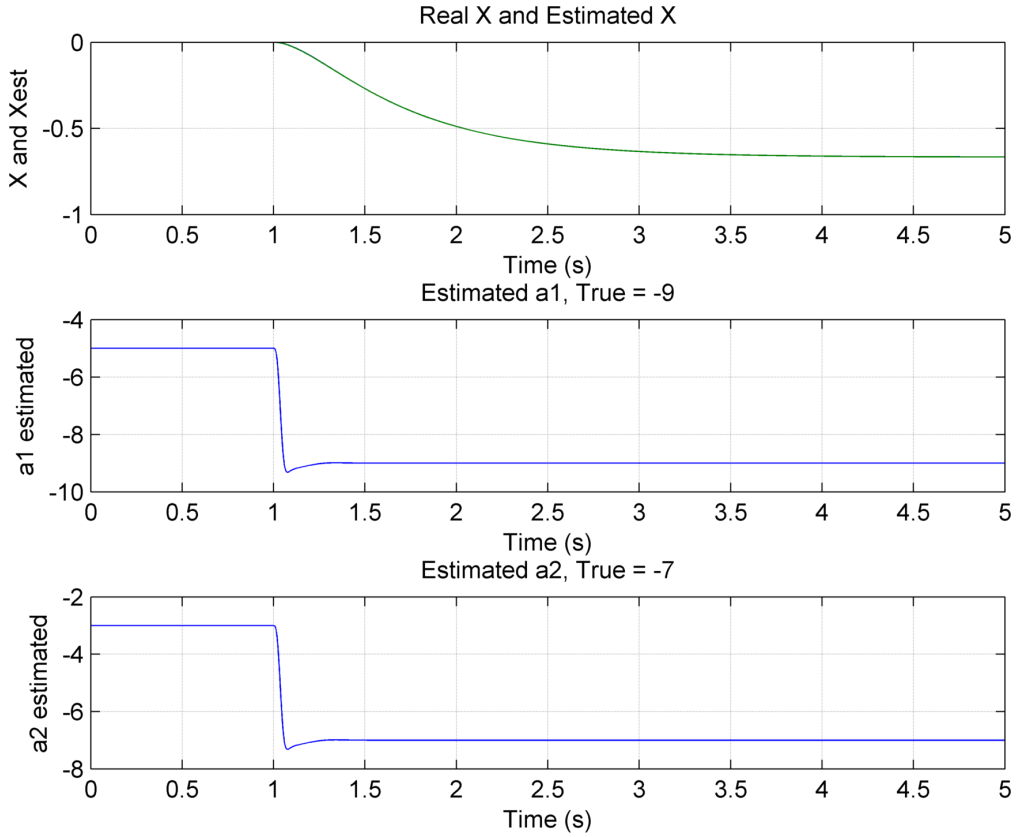

Figure 4.14: Second Order Model Results Case 5, State Tracking and Parameter Convergence

The estimator developed in section 3.3.3 has been demonstrated to successfully determine multiple parameters within a second order system for a variety of different system command inputs and parameters. This provides a solid foundation to move on and investigate a second order non-linear system model with an estimator developed using the same methodology as the second order estimator.

4.1.4 Second Order Nonlinear Model Results

The second order nonlinear model defined in Section 2.1.4 along with the second order nonlinear estimator derived in Section 3.3.4 have been simulated together to ensure proper convergence of the estimated $\hat{a}_1$ to the actual a_1 and the estimated $\hat{a}_2$ to the actual a_2. This results section contains plots illustrating the convergence of the estimated parameters along with the tracking of the model reference state X with the estimated state X_{est}, for step inputs to the system of various magnitudes and a variety of different model parameters.

Table 4.4 shows the four different test conditions and results that are documented in this section, the variables that can be changed are the input type, the actual value of a_1 and a_2, the initial a_1 and a_2, and λ for all second order model tests, η will be .01. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots contained within this section.

Table 4.4: Second Order Nonlinear Model Test Case Table

Test Case No	Command	a_1	a_2	a_{1est}	a_{2est}	λ	a_1Conv	a_2Conv
1	Step, Mag=5	12	8	10	4.2	1000	3 sec	3.5 sec
2	Step, Mag=20	13	10	7	1.8	1000	1.5 sec	1.5 sec
3	Step, Mag=20	25	16	19	8	5000	0.1 sec	0.1 sec
4	Step, Mag=20	-12	-8	-10	-4.2	1000	2 sec	2 sec

Test Case 1

Figure 4.15 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 12, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 8. The plots show that the estimator successfully tracks the system state and takes approximately three seconds to converge upon the actual parameter a_1 and it takes a little over three seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly estimated. This test case illustrates the estimators ability to correctly identify multiple parameters in a second order nonlinear system with a system input of a step command.

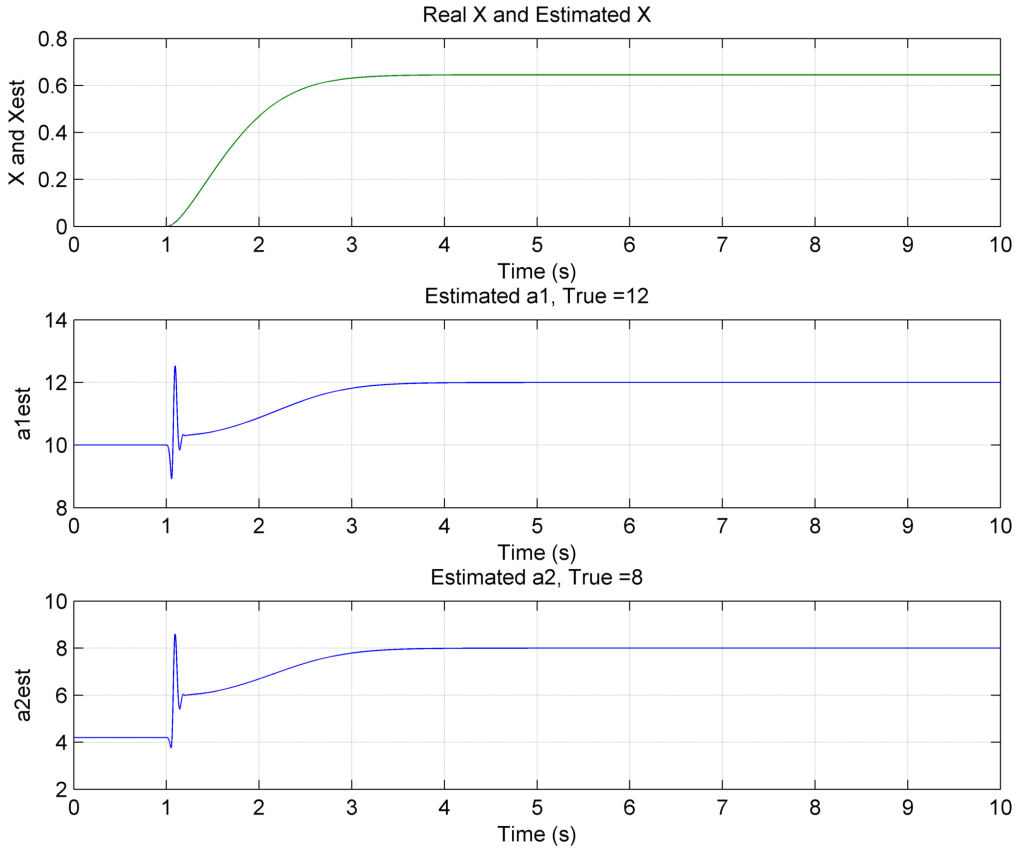

Figure 4.15: Second Order Nonlinear Model Results Case 1, State Tracking and Parameter Convergence

Test Case 2

Figure 4.16 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 13, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 10. The plots show that the estimator successfully tracks the system state and takes approximately one and a half seconds to converge upon the actual parameter a_1 and it takes approximately one and a half seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly estimated. In this test case the magnitude of the step command was increased significantly, resulting in an expected decrease in convergence time. This case also illustrates the ability of the estimator to successfully converge upon the actual parameters when the starting initial a_{1est} and a_{2est} are a significant distance away from

the actual values.

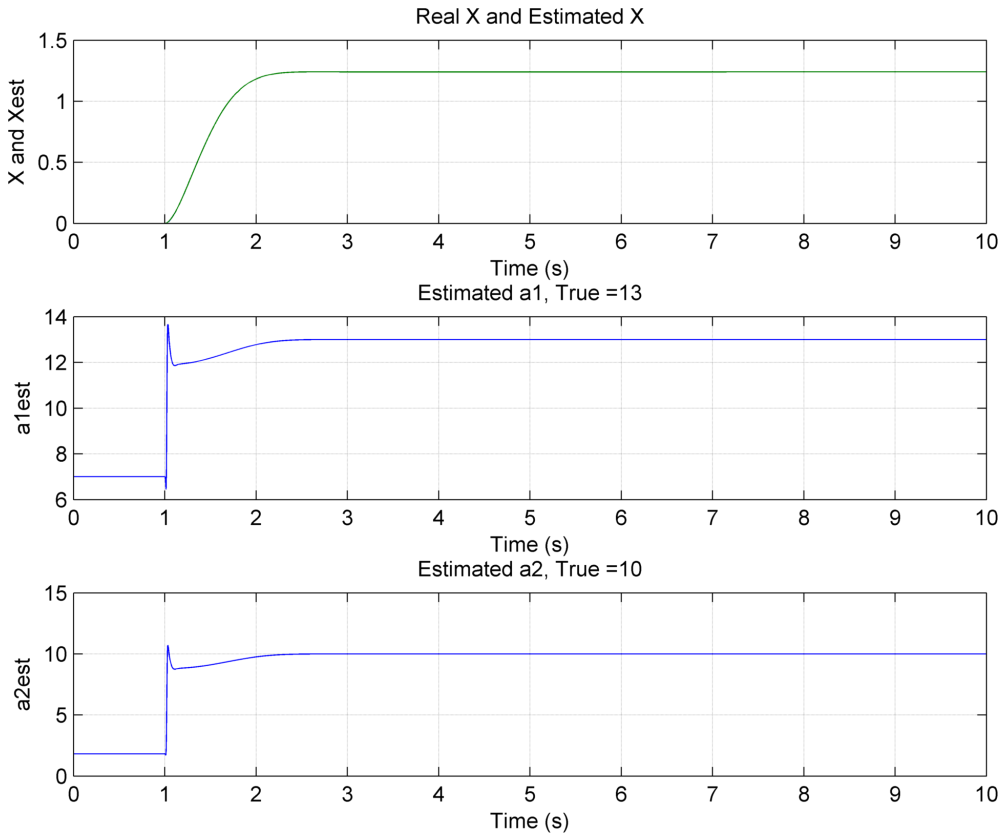

Figure 4.16: Second Order Nonlinear Model Results Case 2, State Tracking and Parameter Convergence

Test Case 3

Figure 4.17 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is 25, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 16. The plots show that the estimator successfully tracks the system state and takes approximately a tenth of a second to converge upon the actual parameter a_1 and it takes approximately one tenth of a second to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after the estimator has converged and the two parameters have been properly estimated. An increase in the λ parameter along with the step input magnitude the same as in test case two result in the fastest convergence time for this system model.

This scenario excites this system very rapidly which leads to fast convergence and also shows the estimator will work with a reference system with larger coefficient values.

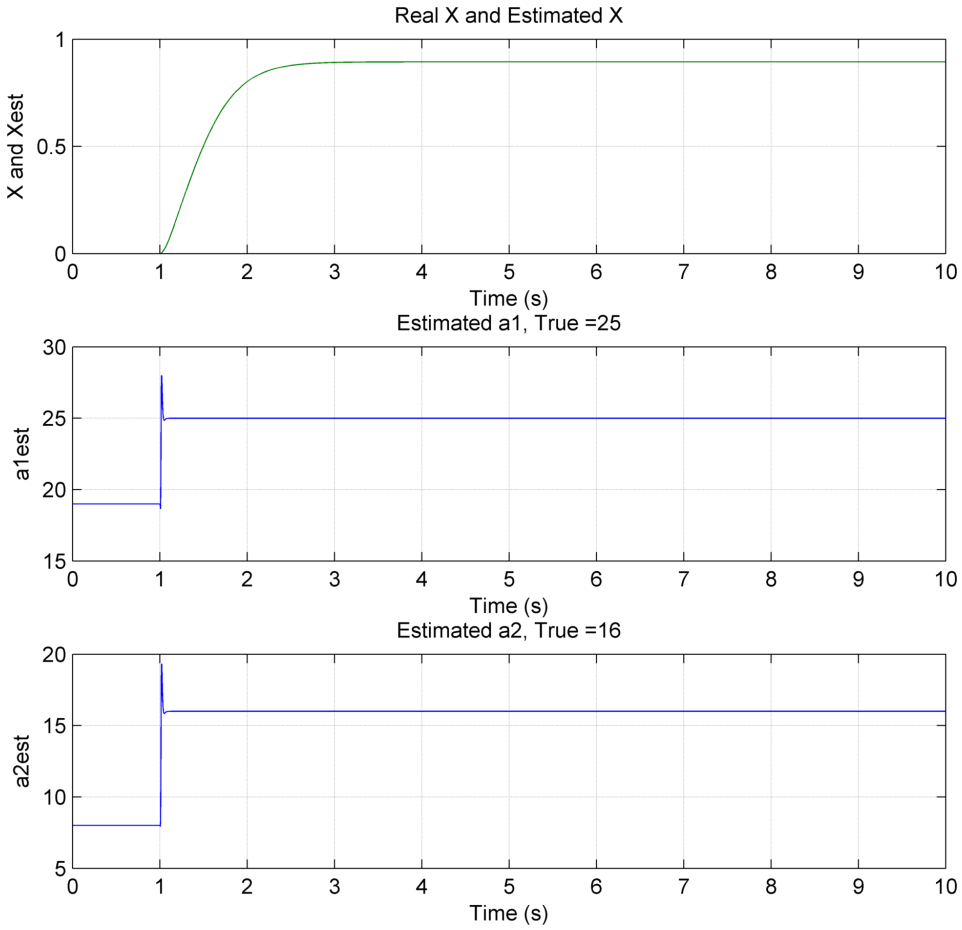

Figure 4.17: Second Order Nonlinear Model Results Case 3, State Tracking and Parameter Convergence

Test Case 4

Figure 4.18 shows three plots, the plot on the top is the tracking of the actual system state X to the estimated system state X_{est}, the plot in the middle shows a_{1est} converging upon the actual value of a_1 which in this case is -12, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is -8. The plots show that the estimator successfully tracks the system state and takes approximately two seconds to converge upon the actual parameter a_1 and it takes a little over two seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X, perfectly after

the estimator has converged and the two parameters have been properly estimated. This test case illustrates the estimators ability to correctly identify multiple parameters in a second order nonlinear system with a system input of a step command in addition to the ability of the parameter estimation routine to properly identify a negative unknown parameter.

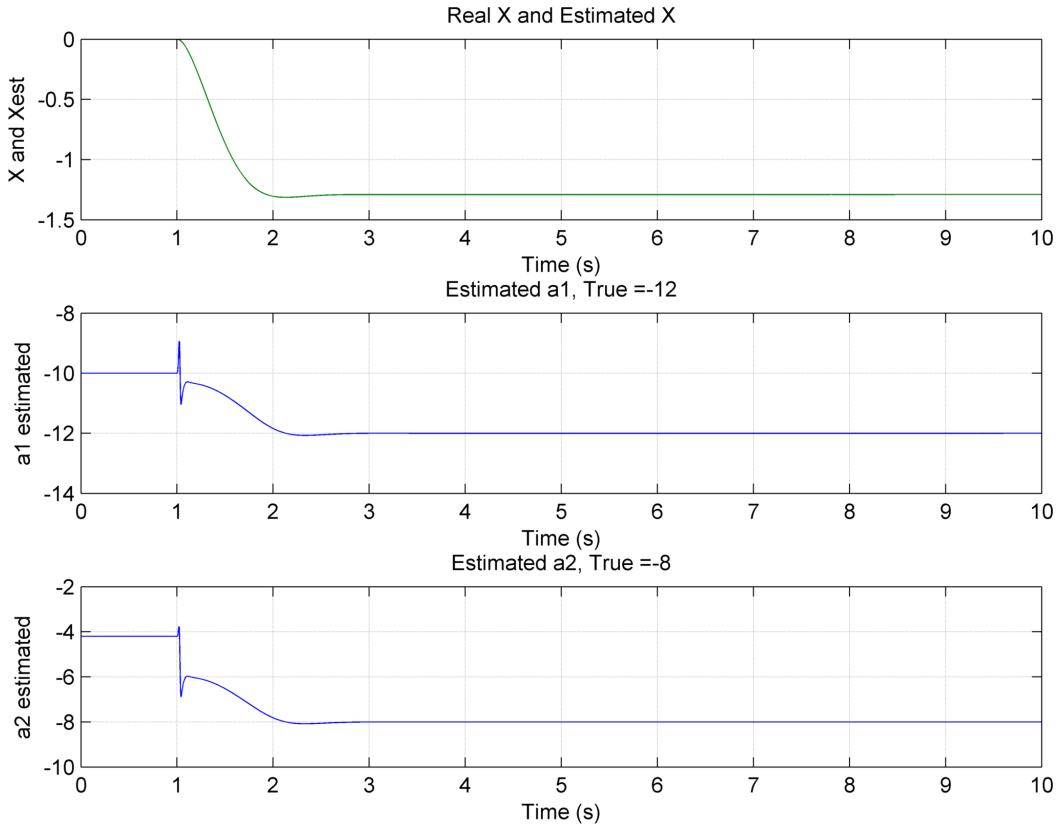

Figure 4.18: Second Order Nonlinear Model Results Case 4, State Tracking and Parameter Convergence

The estimator developed in Section 3.3.4 has been demonstrated to successfully identify multiple parameters within a second order nonlinear system for a variety of different system command inputs and parameters. This provides a solid foundation to move on and investigate a fully nonlinear aircraft model with an estimator developed using the same methodology as the second order nonlinear estimator.

4.1.5 Second Order Nonlinear Sensor Noise Analysis

In this section, sensor noise will be inserted into the second order nonlinear simulation to prove that the estimator is still able to converge under various noise levels. The estimation overview diagram (Figure 3.2) has been modified to include the sensor noise, for this analysis section the simulation will follow the estimation overview shown in Figure 4.19. The band limited white noise will act like sensor noise for the states coming from the reference model, this noise has changed the convergence characteristics of the parameter estimator and the detailed results are shown in this section.

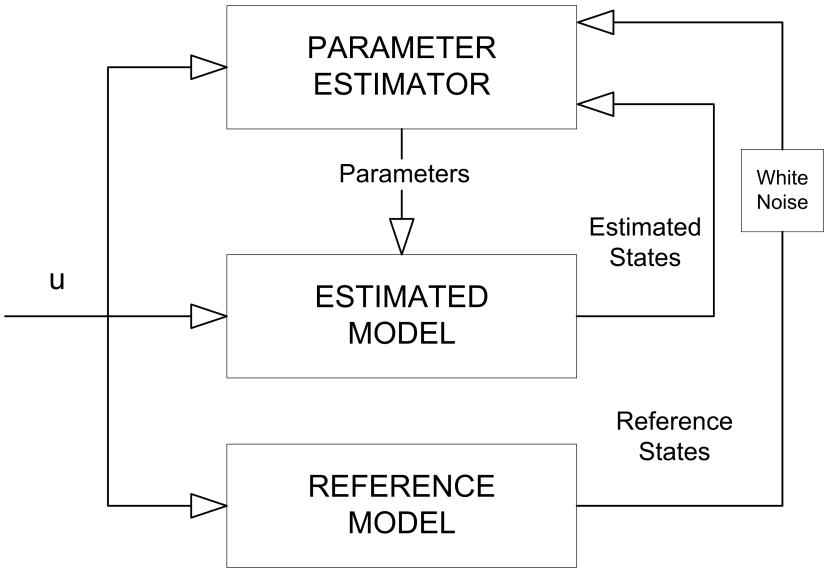

Figure 4.19: Parameter Estimation Overview, Including Noise

The second order nonlinear model defined in Section 2.1.4 along with the second order nonlinear estimator derived in Section 3.3.4 have been simulated together to ensure proper convergence of the estimated $\hat{a}_1$ to the actual a_1 and the estimated $\hat{a}_2$ to the actual a_2 with a system including noise as defined in Figure 4.19. This results section contains plots illustrating the convergence of the estimated parameters along with the tracking of the model reference state X with the estimated state X_{est}, for step inputs to the system of various magnitudes and a variety of different model parameters.

Table 4.6 shows the three different test conditions and results that are documented in this section, the variables that can be changed are the command input type, the white noise gain, and λ for all second order nonlinear model tests, the parameters that remain

constant for all three cases are shown in Table 4.5. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots contained within this section.

Table 4.5: Second Order Nonlinear Model Noise Analysis Constants Table

a_1	a_2	a_{1est}	a_{2est}	η
12	8	10	4	.01

Table 4.6: Second Order Nonlinear Model Noise Analysis Case Table

Test Case No	Command	Noise Gain	λ	a_1Conv	a_2Conv	a_1Final	a_2Final
1	Step, Mag=5	$3.2e^{-3}$	20	3.5 sec	3 sec	11.7722	7.9992
2	Step, Mag=20	$10.0e^{-3}$	3	2 sec	2 sec	11.6216	7.9602
3	Step, Mag=20	$1.0e^{-3}$	30	2 sec	2 sec	11.9585	7.9956

Test Case 1

Figure 4.20 shows four plots, the plot on the top is the tracking of the actual system state X_m to the estimated system state X_{est}, the second plot is the state tracking error (x_{est}-x_m), the third plot shows a_{1est} converging upon the actual value of a_1 which in this case is 12, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 8. The plots show that the estimator successfully tracks the system state and takes approximately three and a half seconds to converge upon the actual parameter a_1 and it takes a little over three seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X_m, correctly after the estimator has converged and the two parameters have been properly estimated. The noise in the system prevents the estimator to identify the exact parameter for both a_1 and a_2, the best parameter estimate will occur with the lowest noise level.

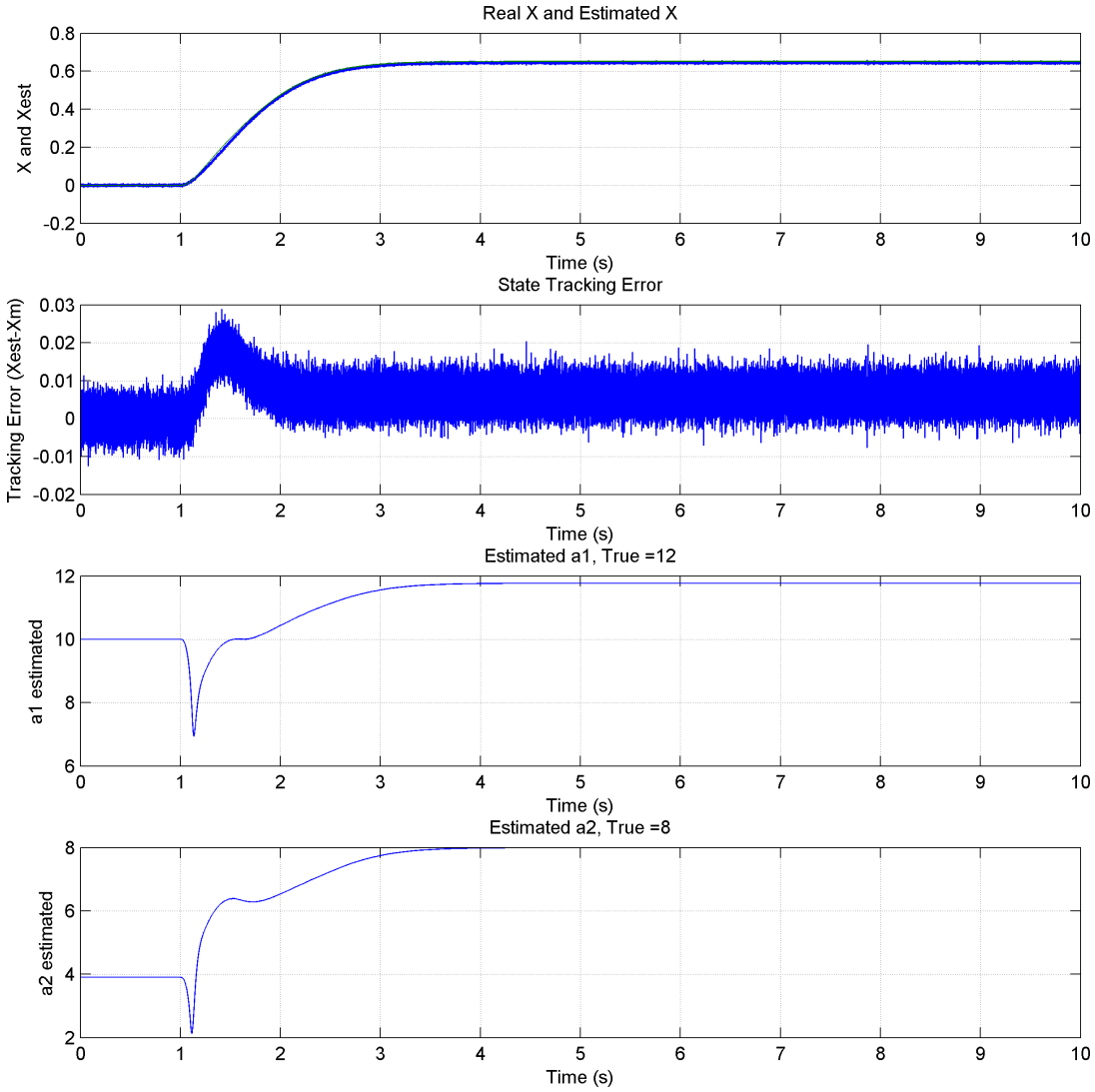

Figure 4.20: Second Order Nonlinear Model Noise Analysis Case 1, State Tracking and Parameter Convergence

Test Case 2

Figure 4.21 shows four plots, the plot on the top is the tracking of the actual system state X_m to the estimated system state X_{est}, the second plot is the state tracking error (x_{est}-x_m), the third plot shows a_{1est} converging upon the actual value of a_1 which in this case is 12, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 8. The plots show that the estimator successfully tracks the system state and takes approximately two seconds to converge upon the actual parameter a_1 and it takes approximately two seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X_m, correctly after the estimator has converged and the two parameters have been properly estimated. The noise in the system prevents the estimator to identify the exact parameter for both a_1 and a_2, the noise level in this test case number two is the highest level analyzed and results in the worst estimate of the two parameters. The input to the system in this case was changed to a magnitude of 20 step command, this results in a faster convergence time, however the accuracy is a function of the noise in the system and is not improved with a higher magnitude step command input.

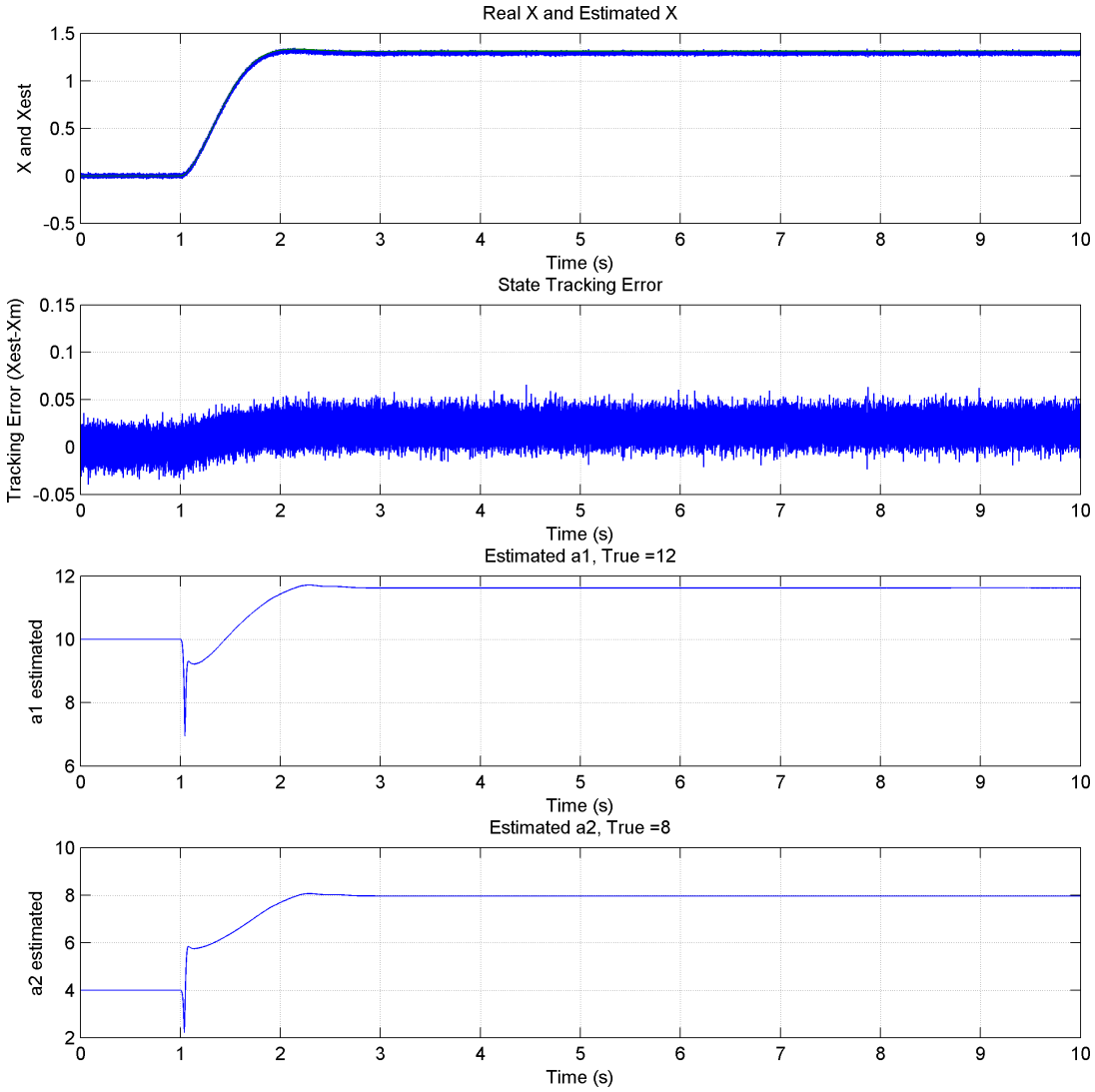

Figure 4.21: Second Order Nonlinear Model Noise Analysis Case 2, State Tracking and Parameter Convergence

Test Case 3

Figure 4.22 shows four plots, the plot on the top is the tracking of the actual system state X_m to the estimated system state X_{est}, the second plot is the state tracking error (x_{est}-x_m), the third plot shows a_{1est} converging upon the actual value of a_1 which in this case is 12, and the plot on the bottom shows a_{2est} converging upon the actual value of a_2 which in this case is 8. The plots show that the estimator successfully tracks the system state and takes approximately two seconds to converge upon the actual parameter a_1 and it takes approximately two seconds to converge on to a_2. The resulting estimated state, X_{est}, tracks the actual state, X_m, correctly after the estimator has converged and the two parameters have been properly estimated. The noise in the system prevents the estimator to identify the exact parameter for both a_1 and a_2, the noise level in this test case number three is the lowest level analyzed and results in the best estimate of the two parameters. The input to the system in this case was changed to a magnitude of 20 step command, this results in a faster convergence time, of the three noise analysis test cases, this one provides the fastest convergence time and the highest level of accuracy due to the low noise level.

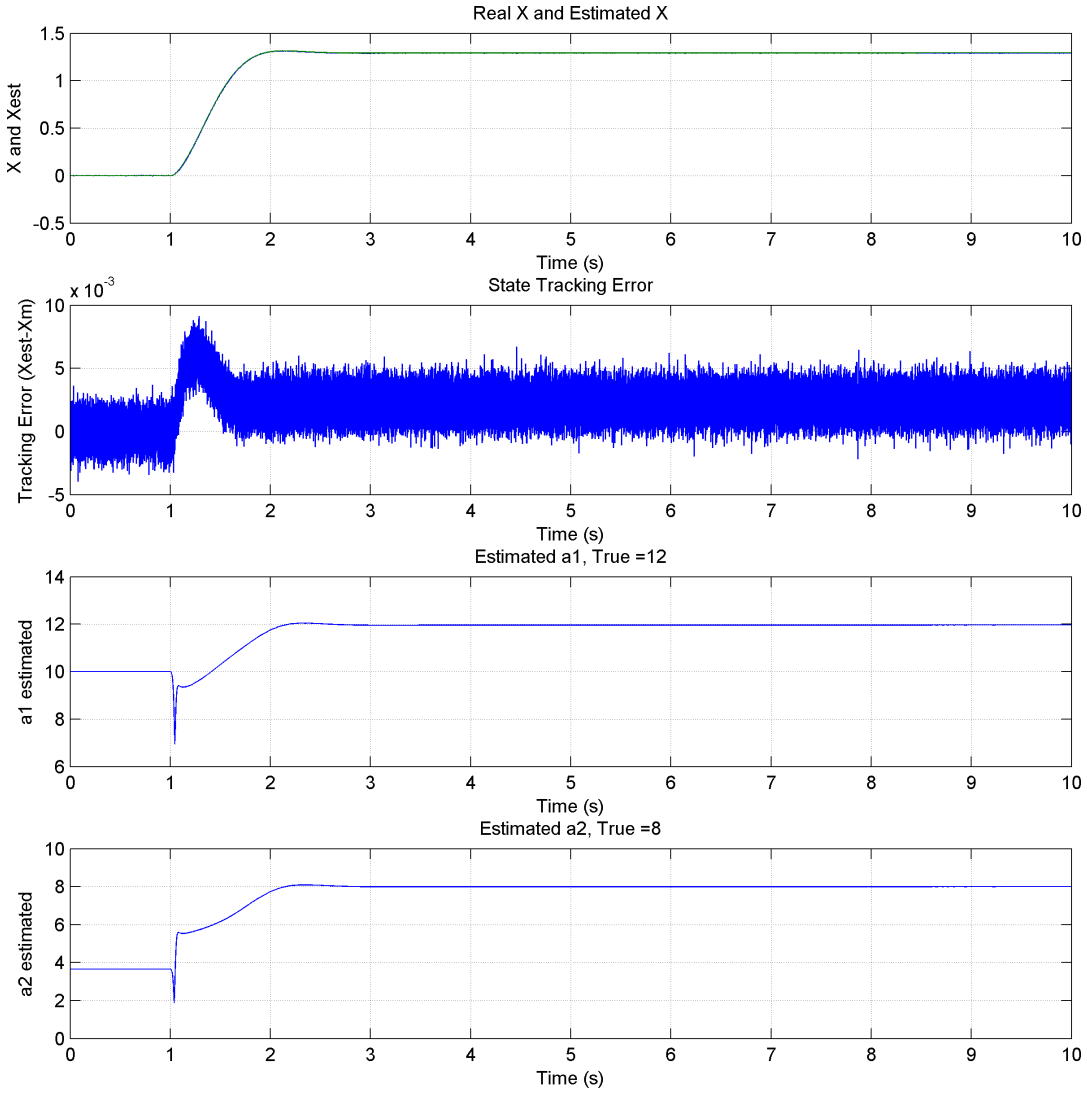

Figure 4.22: Second Order Nonlinear Model Noise Analysis Case 3, State Tracking and Parameter Convergence

The estimator developed in Section 3.3.4 has been demonstrated to successfully identify multiple parameters within a second order nonlinear system that includes sensor noise for a variety of different system command inputs and parameters. The accuracy of the parameter estimation has been found to be directly related to the sensor noise levels found in the system.

4.2 Aircraft Model Results

There are three parts to the MATLAB and Sumulink analysis that is performed in order to estimate aerodynamic coefficients. The first part of the system is the model reference aircraft model, this model is simulating the actual aircraft that the parameter estimator would be running on. The model reference provides the measurable state information that is required by the estimator. The second part of the system is the estimator itself, the estimator equations are shown in Section 3.4.2. The estimated aerodynamic coefficient is then fed into another aircraft model that used the estimated aerodynamic coefficients instead of the actual coefficients. This setup creates a simulated real-time environment for the parameter estimation to take place, and allows for the detailed comparison of model reference aircraft states to the estimated aircraft states. An overview diagram of the simulation is shown in Figure 3.2. The results of this simulation are shown in this section and the detailed model is documented in Appendix A.

4.2.1 Longitudinal PID

C_m

The aircraft model defined in Section 2.2 and the aircraft pitching moment coefficient, C_m, estimators derived in Section 3.4.1 have been simulated together to verify proper convergence of the estimated parameters, ($\Delta\hat{C}_{m_\alpha}$ and $\Delta\hat{C}_{m_q}$) to the actual parameters, (ΔC_{m_α} and ΔC_{m_q}). This results section contains plots illustrating the convergence of the estimated parameters along with the tracking of the model reference aircraft states to the estimated states for both of the C_m unknown parameters.

$\Delta\hat{C}_{m_\alpha}$ Results

The change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$, estimator derived in Section 3.4.1 has been programmed and simulated with the results shown in this section. Table 4.7 shows the four different test conditions and results that are documented in this section, the variables that can be changed are the aircraft control command, the actual value of $\hat{C}_{m_\alpha}$, the upper and lower $\hat{C}_{m_\alpha}$. For all of the $\Delta\hat{C}_{m_\alpha}$ test cases, $\lambda = 10^{-7}$ and $\eta = 10^{-3}$. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots below. This section contains results from running the $\Delta\hat{C}_{m_\alpha}$ estimator only, multiple estimators have been run at the same time and the results can be found in Section 4.2.1.

Table 4.7: $\hat{C}_{m_\alpha}$ Test Case Table

Test Case	dhs Command	C_{m_α}	$C_{m_{\alpha_{upp}}}$	$C_{m_{\alpha_{low}}}$	Final Value	Est Time
1	Doublet, Mag=-.05deg	-.0015	-.001	-.01	$-1.5177e^{-3}$	.5 Sec
2	Doublet, Mag=-.05deg	-.0015	-.0001	-.01	$-1.4946e^{-3}$	.5 Sec
3	Doublet, Mag=-.05deg	-.0011	-.001	-.01	$-1.1297e^{-3}$	4 Sec
4	Sin=-.05deg Fr=1Hz	-.0015	-.0001	-.008	$-1.0966e^{-3}$	.4 Sec

$\Delta\hat{C}_{m_\alpha}$ Test Case 1

Figure 4.23 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a $-.05$ degree doublet maneuver to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

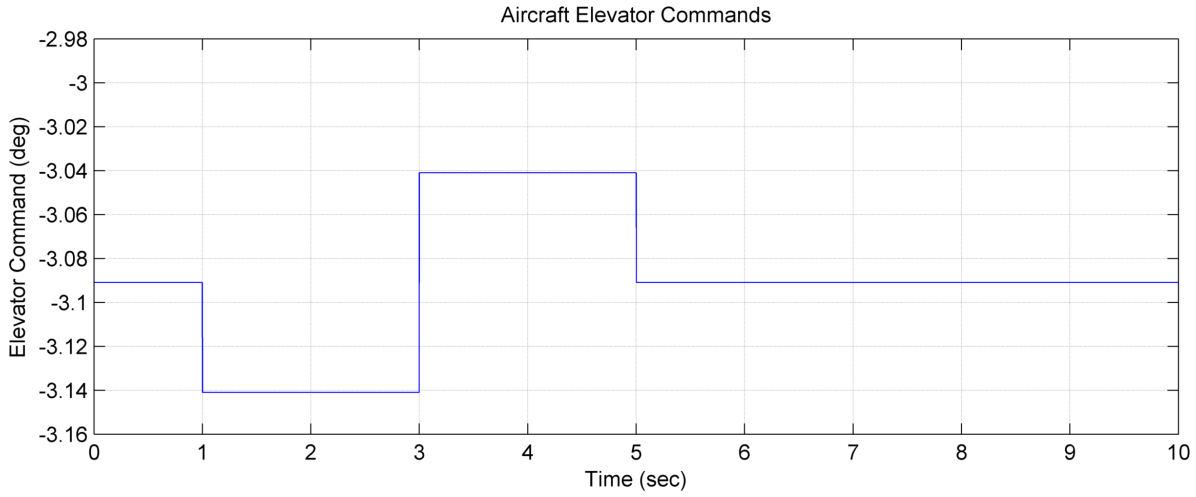

Figure 4.23: $\Delta\hat{C}_{m_\alpha}$ Test Case 1, PID Aircraft Control Commands

Figure 4.24 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_\alpha}$ to the actual parameter $\Delta C_{m_\alpha} = -.0015$, the system takes approximately half a second to converge upon the model reference parameter.

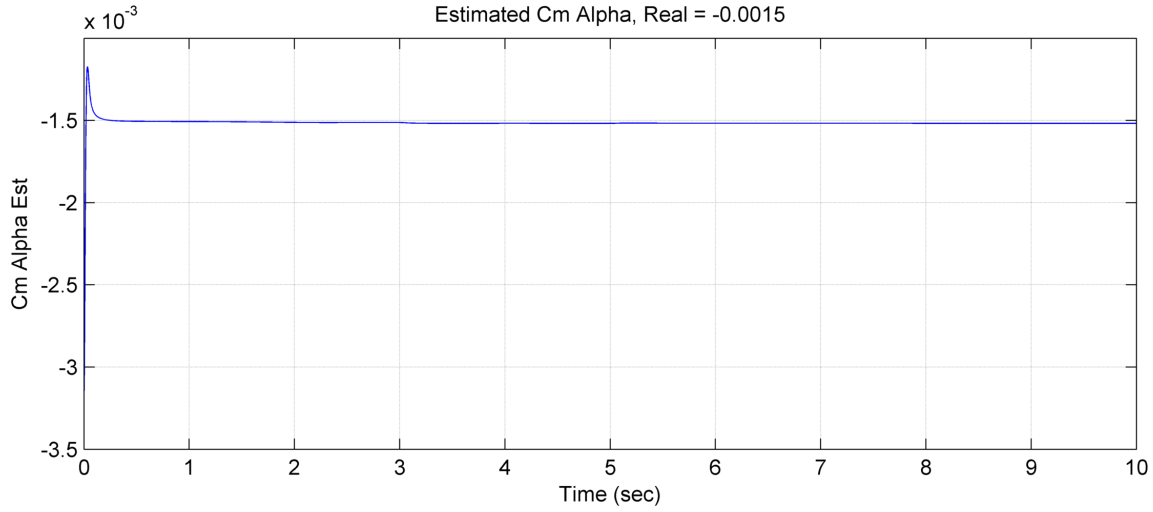

Figure 4.24: $\Delta\hat{C}_{m_\alpha}$ Test Case 1, Converging Upon ΔC_{m_α}

Figure 4.25 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_\alpha}$ parameter has converged upon the actual ΔC_{m_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

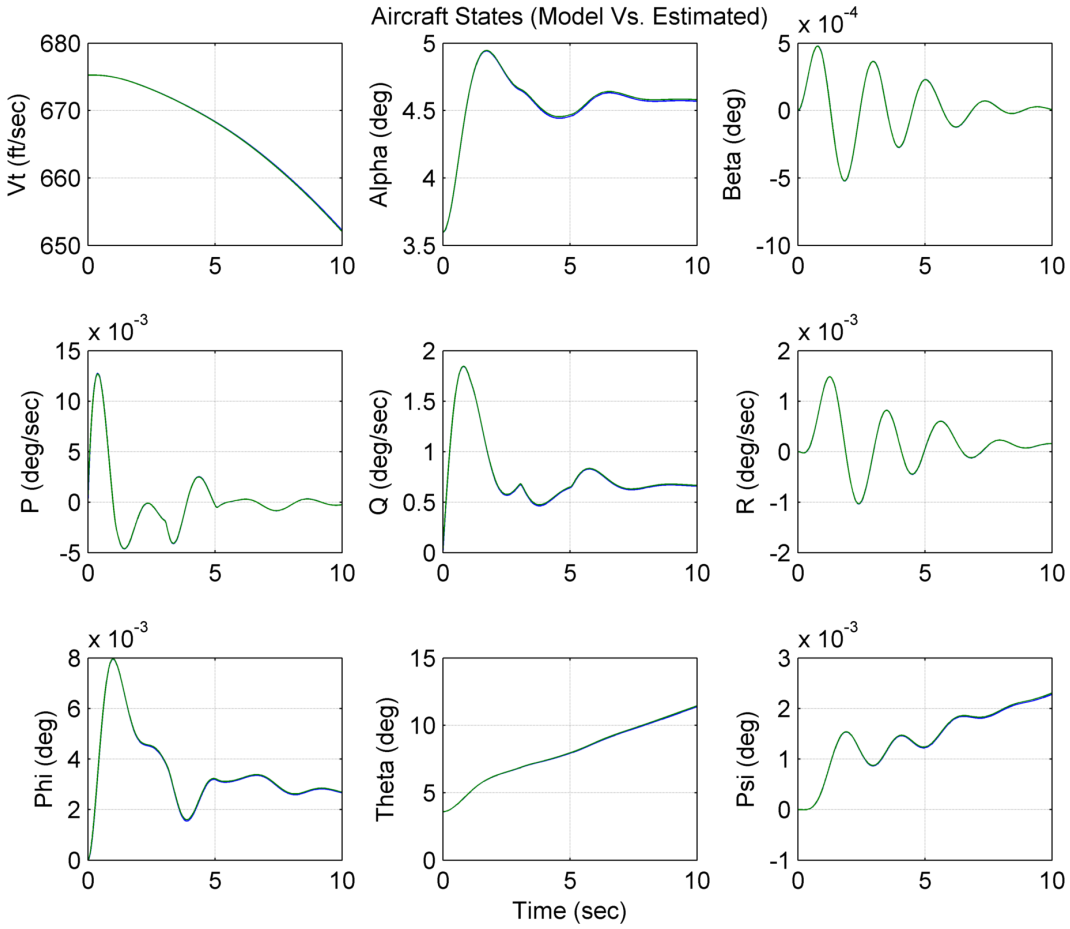

Figure 4.25: $\Delta\hat{C}_{m_\alpha}$ Test Case 1, Aircraft State Tracking

$\Delta\hat{C}_{m_\alpha}$ Test Case 2

Figure 4.26 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a $-.05$ degree doublet maneuver to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

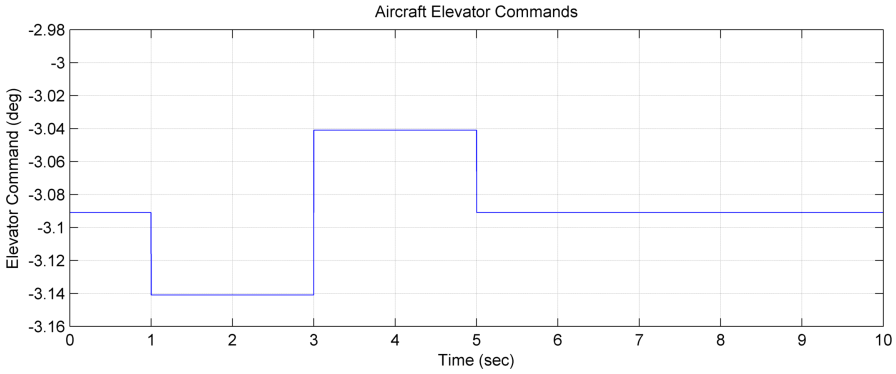

Figure 4.26: $\Delta\hat{C}_{m_\alpha}$ Test Case 2, PID Aircraft Control Commands

Figure 4.27 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_\alpha}$ to the actual parameter $\Delta C_{m_\alpha} = -.0015$, the system takes approximately half a second to converge upon the model reference parameter. In this test case, the initial starting parameter is higher than the actual parameter, this directional change is different from the test case 1, this test case is illustrating the estimator's ability to change the upper and lower limits on the initial parameter bounds while still converging on the actual parameter.

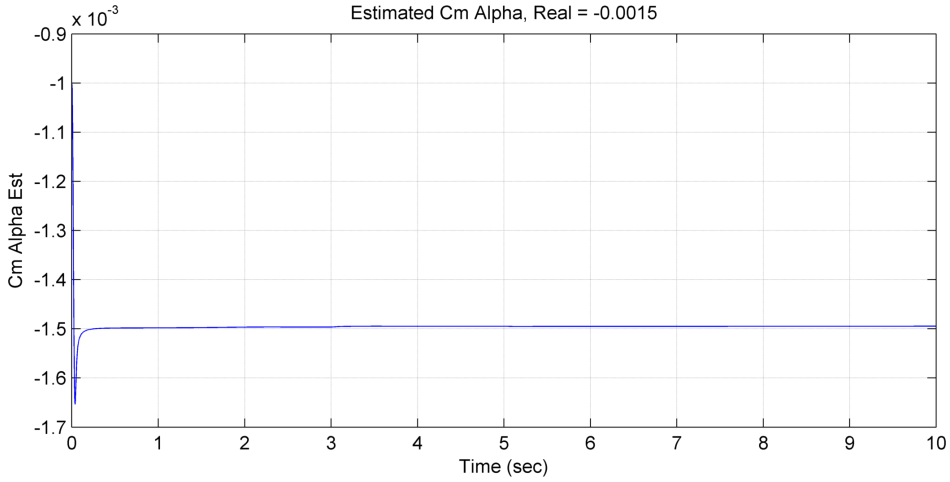

Figure 4.27: $\Delta\hat{C}_{m_\alpha}$ Test Case 2, Converging Upon ΔC_{m_α}

Figure 4.28 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_\alpha}$ parameter has converged upon the actual ΔC_{m_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

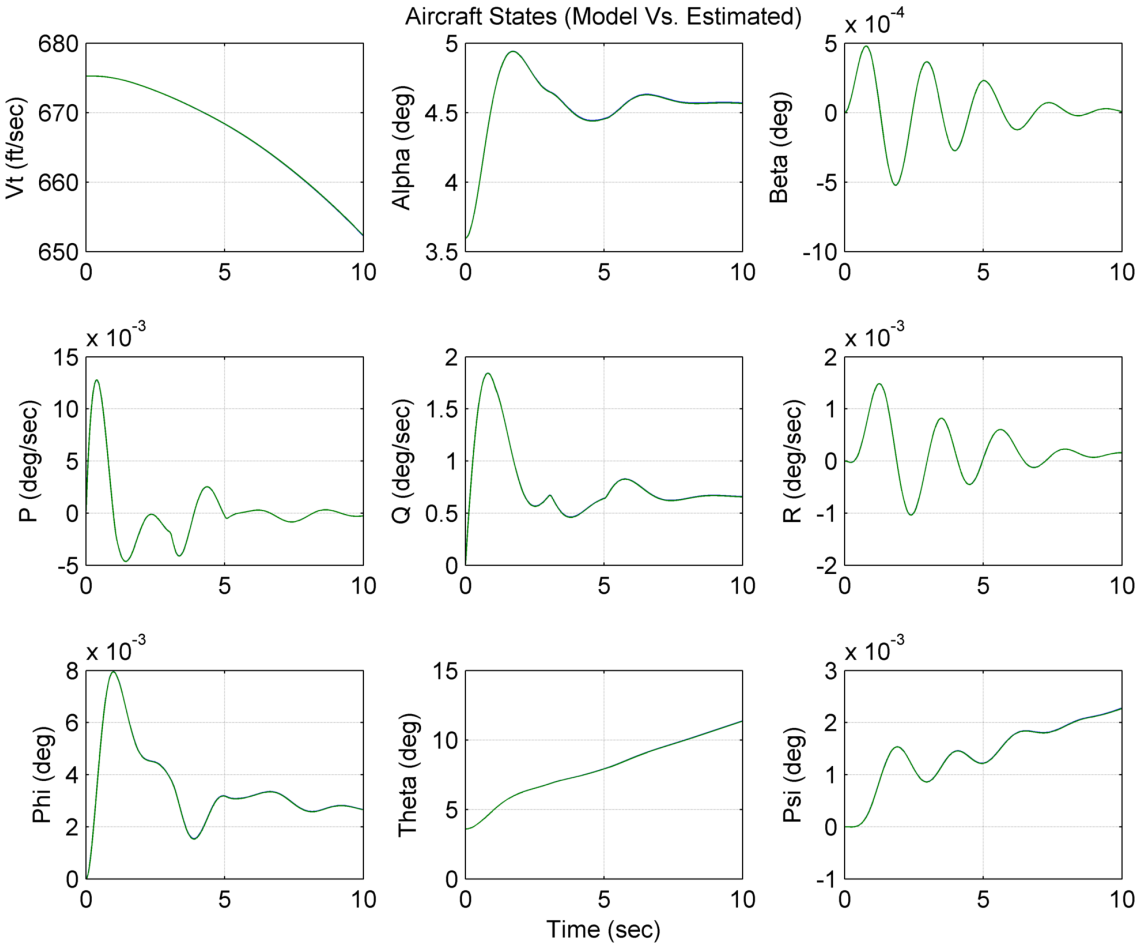

Figure 4.28: $\Delta\hat{C}_{m_\alpha}$ Test Case 2, Aircraft State Tracking

$\Delta\hat{C}_{m_\alpha}$ Test Case 3

Figure 4.29 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a $-.05$ degree doublet maneuver to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

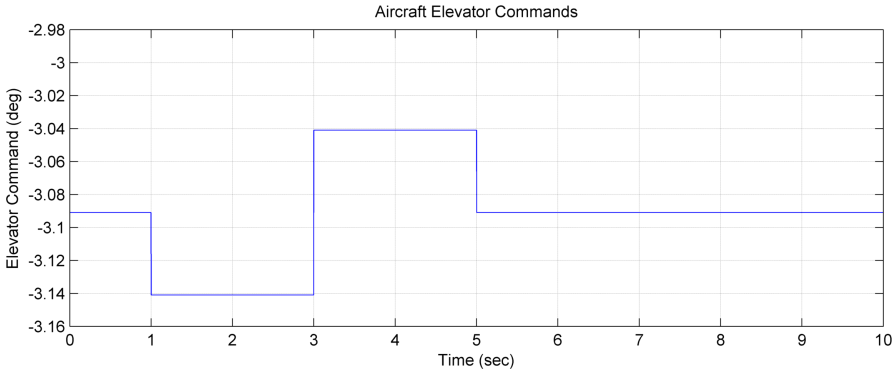

Figure 4.29: $\Delta\hat{C}_{m_\alpha}$ Test Case 3, PID Aircraft Control Commands

Figure 4.30 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_\alpha}$ to the actual parameter $\Delta C_{m_\alpha} = -.0011$, the system takes approximately four seconds to converge upon the model reference parameter. In this test case, the actual parameter has been changed while leaving the upper and lower bounds constant. This shows that when the initial guess is further away from the actual parameter the time to converge is increased. This is an expected result and this test case verified this claim.

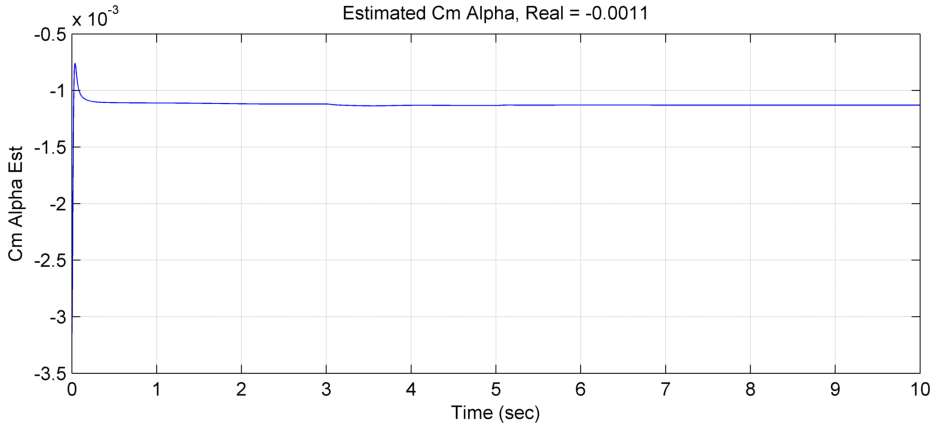

Figure 4.30: $\Delta\hat{C}_{m_\alpha}$ Test Case 3, Converging Upon ΔC_{m_α}

Figure 4.31 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_\alpha}$ parameter has converged upon the actual ΔC_{m_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

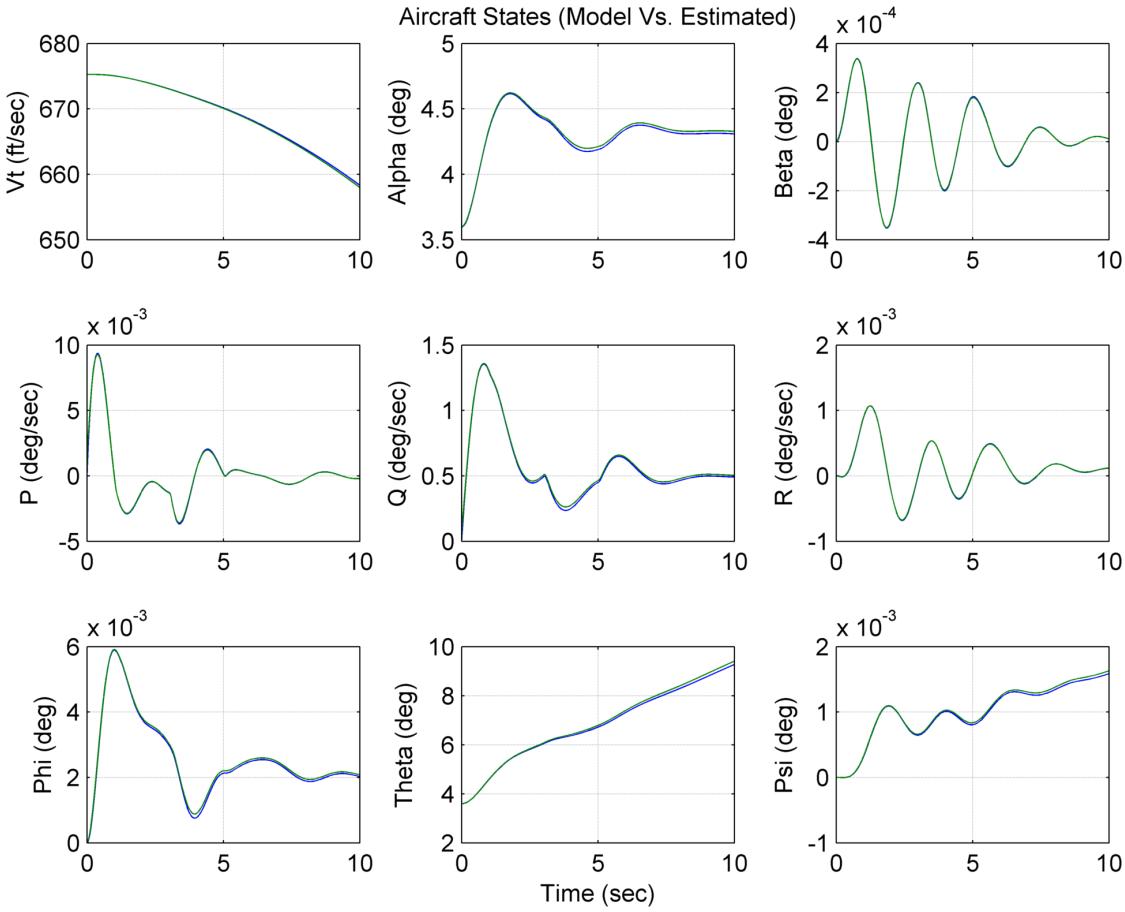

Figure 4.31: $\Delta\hat{C}_{m_\alpha}$ Test Case 3, Aircraft State Tracking

$\Delta\hat{C}_{m_\alpha}$ Test Case 4

Figure 4.32 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .05 degree, 1Hz sin wave to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters. This command is intended to simulate a pilot input to the system at a frequency of 1 Hz.

Figure 4.32: $\Delta\hat{C}_{m_\alpha}$ Test Case 4, PID Aircraft Control Commands

Figure 4.33 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_\alpha}$ to the actual parameter $\Delta C_{m_\alpha} = -.0015$, the system takes approximately half a second to converge upon the model reference parameter. In this test case the system is excited with a sin wave instead of a simple doublet command, this offers a higher level of excitation and results in the fastest and most accurate convergence.

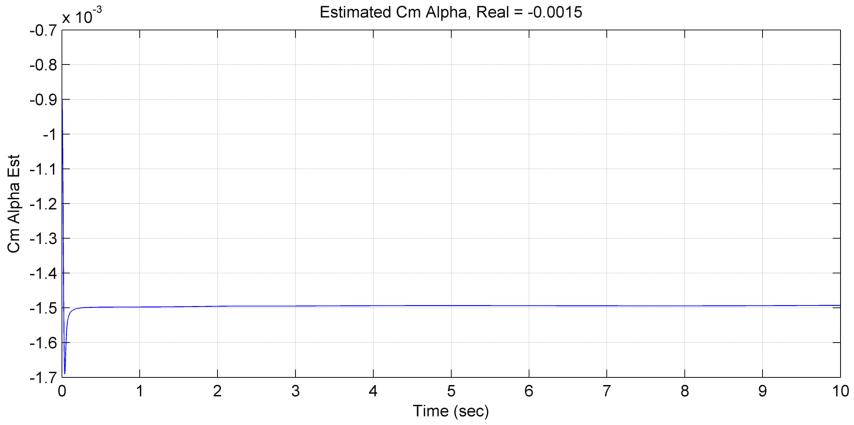

Figure 4.33: $\Delta\hat{C}_{m_\alpha}$ Test Case 4, Converging Upon ΔC_{m_α}

Figure 4.34 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_\alpha}$ parameter has converged upon the actual ΔC_{m_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

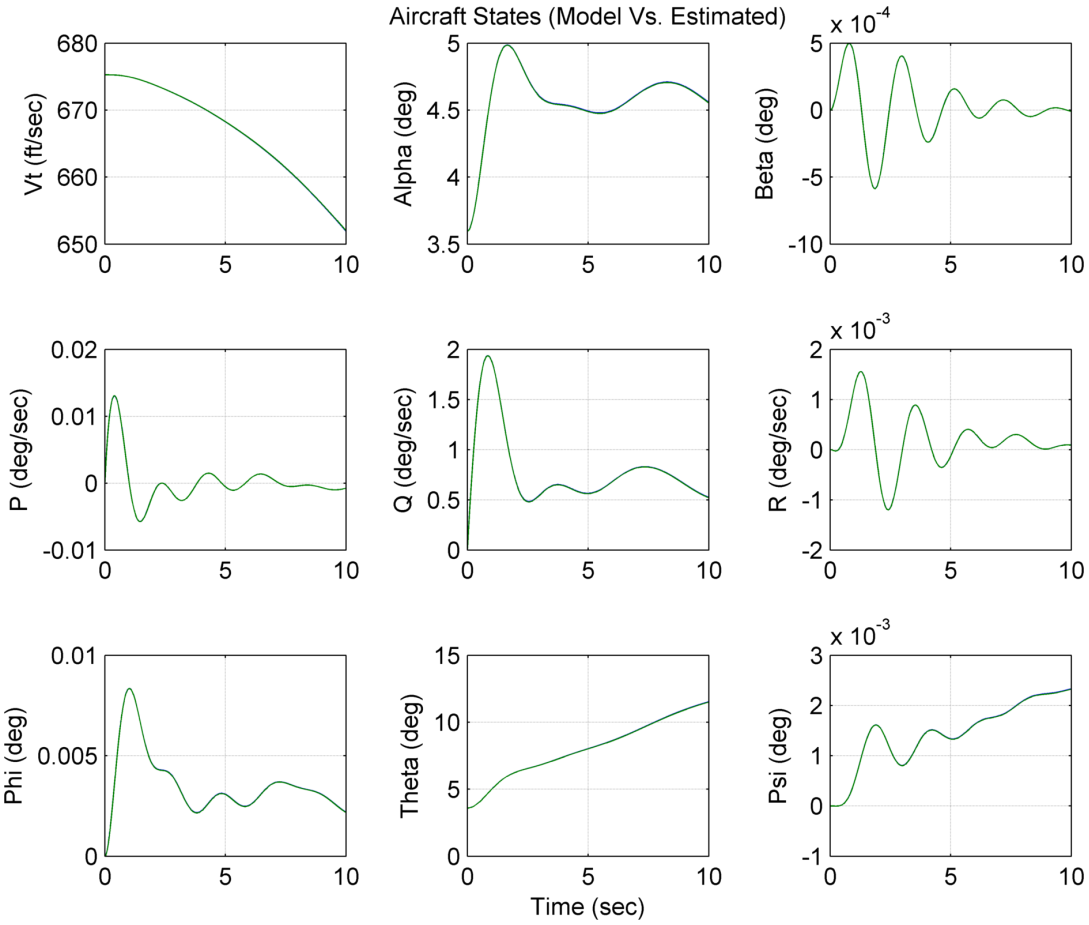

Figure 4.34: $\Delta\hat{C}_{m_\alpha}$ Test Case 4, Aircraft State Tracking

$\Delta\hat{C}_{m_q}$ Results

The change in pitching moment due to a change in pitch rate, $\Delta\hat{C}_{m_q}$, estimator derived in Section 3.4.1 has been programmed and simulated with the results shown in this section. Table 4.8 shows the three different test conditions and results that are documented in this section, the variables that can be changed are the aircraft control command, the actual value of $\hat{C}_{m_q}$, η, and the upper and lower bounds of $\hat{C}_{m_q}$. For all of the $\Delta\hat{C}_{m_q}$ test cases, $\lambda = 2^4$. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots below. This section contains results from running the $\Delta\hat{C}_{m_q}$ estimator only, multiple estimators have been run at the same time and the results can be found in Section 4.2.1.

Table 4.8: $\hat{C}_{m_q}$ Test Case Table

Test Case	dhs Command	C_{m_q}	$C_{m_{q_{upp}}}$	$C_{m_{q_{low}}}$	η	Value @ 10 Sec	Est Time
1	Sin, .05deg, 1Hz	.01	.05	.001	.01	$1.0088e^{-2}$	15 Sec
2	Sin, .05deg, 1Hz	.01	.011	.0099	.01	$0.9987e^{-2}$	13 Sec
3	Doublet, -.05deg	.05	.001	.0099	.01	$0.9863e^{-2}$	18 Sec

$\Delta\hat{C}_{m_q}$ Test Case 1

Figure 4.35 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a $-.05$ degree sin wave to the elevator at 1Hz. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

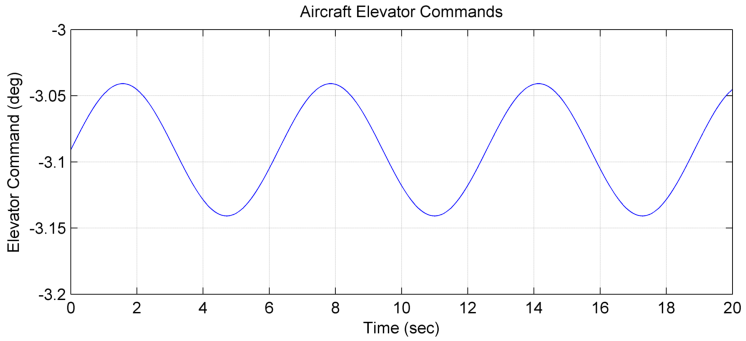

Figure 4.35: $\Delta\hat{C}_{m_q}$ Test Case 1, PID Aircraft Control Commands

Figure 4.36 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_q}$ to the actual parameter $\Delta C_{m_q} = .01$, the system takes approximately fifteen seconds to converge upon the model reference parameter. This time to convergence is significantly more than the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$ estimation. The convergence is stable, and changing the gains and control command excitations will allow for faster convergence times.

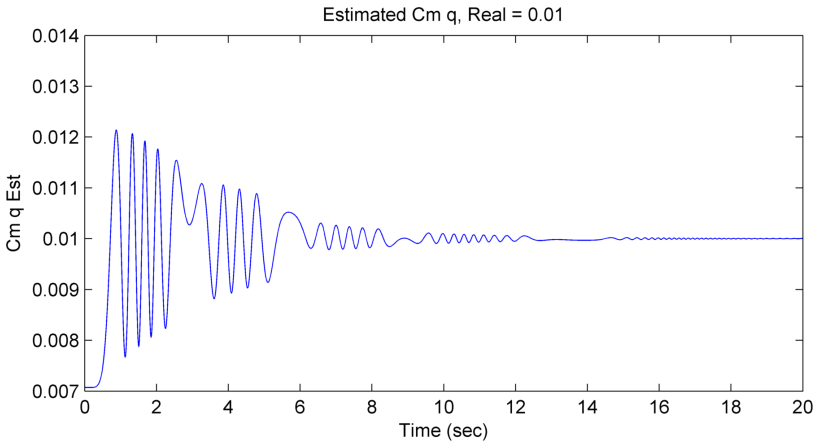

Figure 4.36: $\Delta\hat{C}_{m_q}$ Test Case 1, Converging Upon ΔC_{m_q}

Figure 4.37 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_q}$ parameter has converged upon the actual ΔC_{m_q} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in pitch rate, $\Delta\hat{C}_{m_q}$, successfully estimates the correct parameter and the system states are tracked as expected.

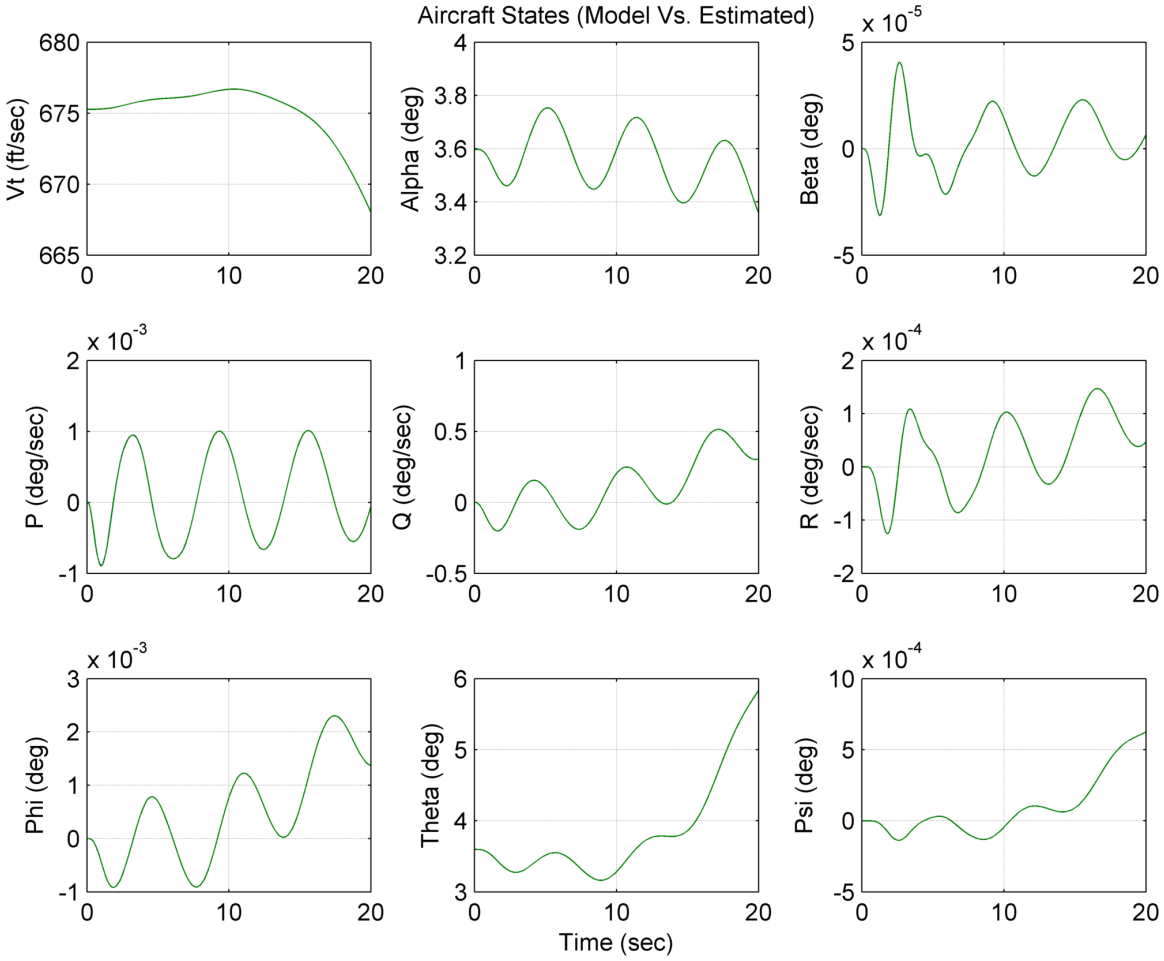

Figure 4.37: $\Delta\hat{C}_{m_q}$ Test Case 1, Aircraft State Tracking

$\Delta\hat{C}_{m_q}$ Test Case 2

Figure 4.38 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a $-.05$ degree sin wave to the elevator at 1Hz. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

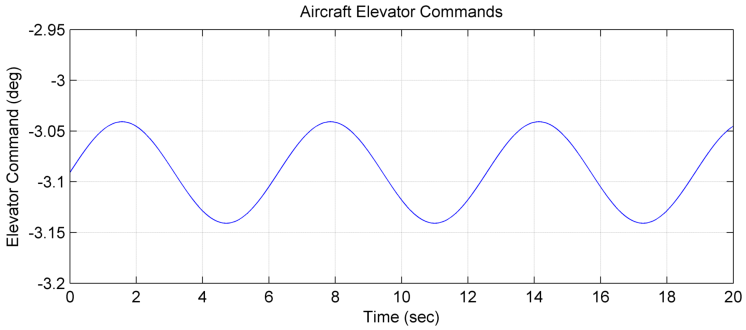

Figure 4.38: $\Delta\hat{C}_{m_q}$ Test Case 2, PID Aircraft Control Commands

Figure 4.39 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_q}$ to the actual parameter $\Delta C_{m_q} = .01$, the system takes approximately thirteen seconds to converge upon the model reference parameter. This time to convergence is significantly more than the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$ estimation. The convergence is stable, and changing the gains and control command excitations will allow for faster convergence times. The reduced range of parameter bounds in this case has decreased the convergence time from test case 1.

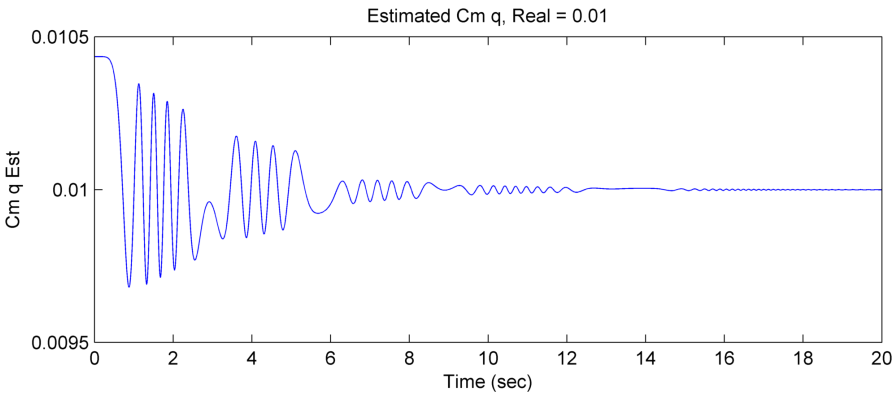

Figure 4.39: $\Delta\hat{C}_{m_q}$ Test Case 2, Converging Upon ΔC_{m_q}

Figure 4.40 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_q}$ parameter has converged upon the actual ΔC_{m_q} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in pitch rate, $\Delta\hat{C}_{m_q}$, successfully estimates the correct parameter and the system states are tracked as expected.

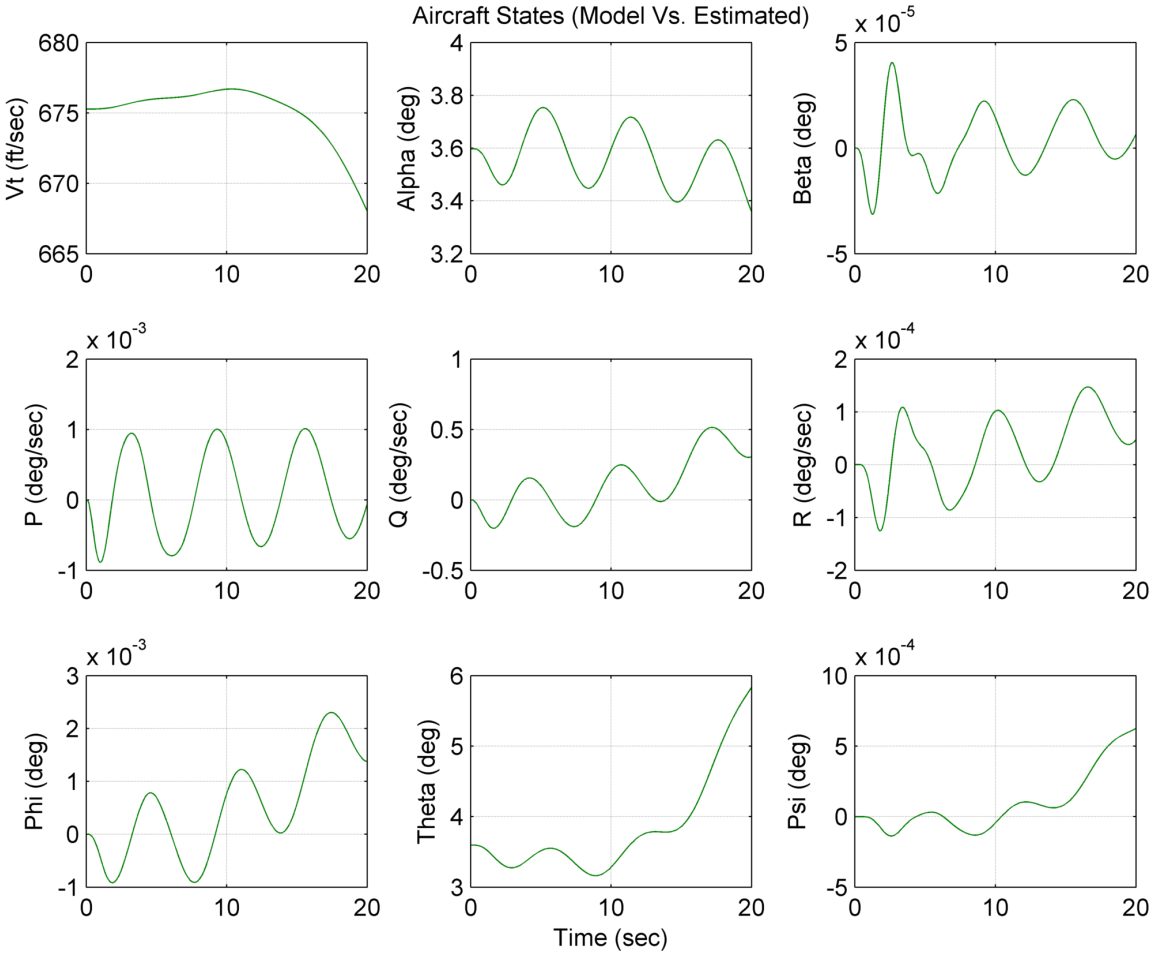

Figure 4.40: $\Delta\hat{C}_{m_q}$ Test Case 2, Aircraft State Tracking

$\Delta\hat{C}_{m_q}$ Test Case 3

Figure 4.41 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a $-.05$ degree doublet command. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

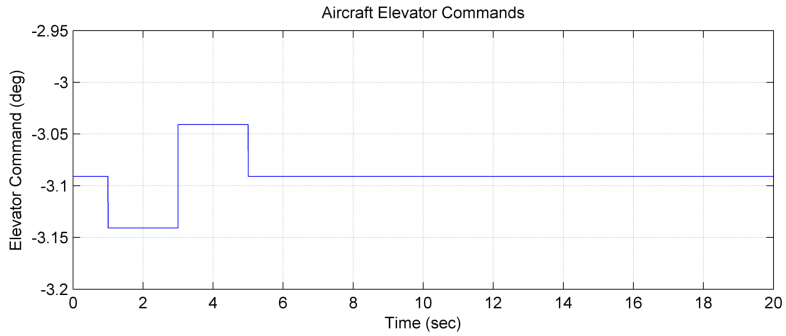

Figure 4.41: $\Delta\hat{C}_{m_q}$ Test Case 3, PID Aircraft Control Commands

Figure 4.42 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_q}$ to the actual parameter $\Delta C_{m_q} = .01$, the system takes approximately eighteen seconds to converge upon the model reference parameter. This time to convergence is significantly more than the change in pitching moment due to a change in angle of attack, $\Delta\hat{C}_{m_\alpha}$ estimation. The convergence is stable, and changing the gains and control command excitations will allow for faster convergence times. The reduced input excitation from the sin wave in case 1 and 2 has increased the amount of time it takes for the parameter to converge. The reduction in η value has led to much more transient jumps and a more agressive chattering.

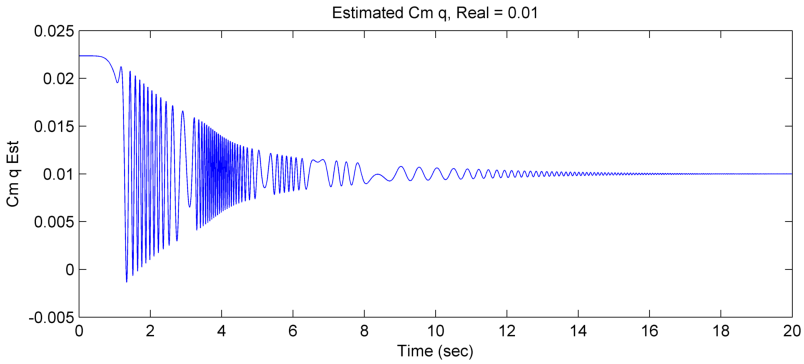

Figure 4.42: $\Delta\hat{C}_{m_q}$ Test Case 3, Converging Upon ΔC_{m_q}

Figure 4.43 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{m_q}$ parameter has converged upon the actual ΔC_{m_q} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in pitching moment due to a change in pitch rate, $\Delta\hat{C}_{m_q}$, successfully estimates the correct parameter and the system states are tracked as expected.

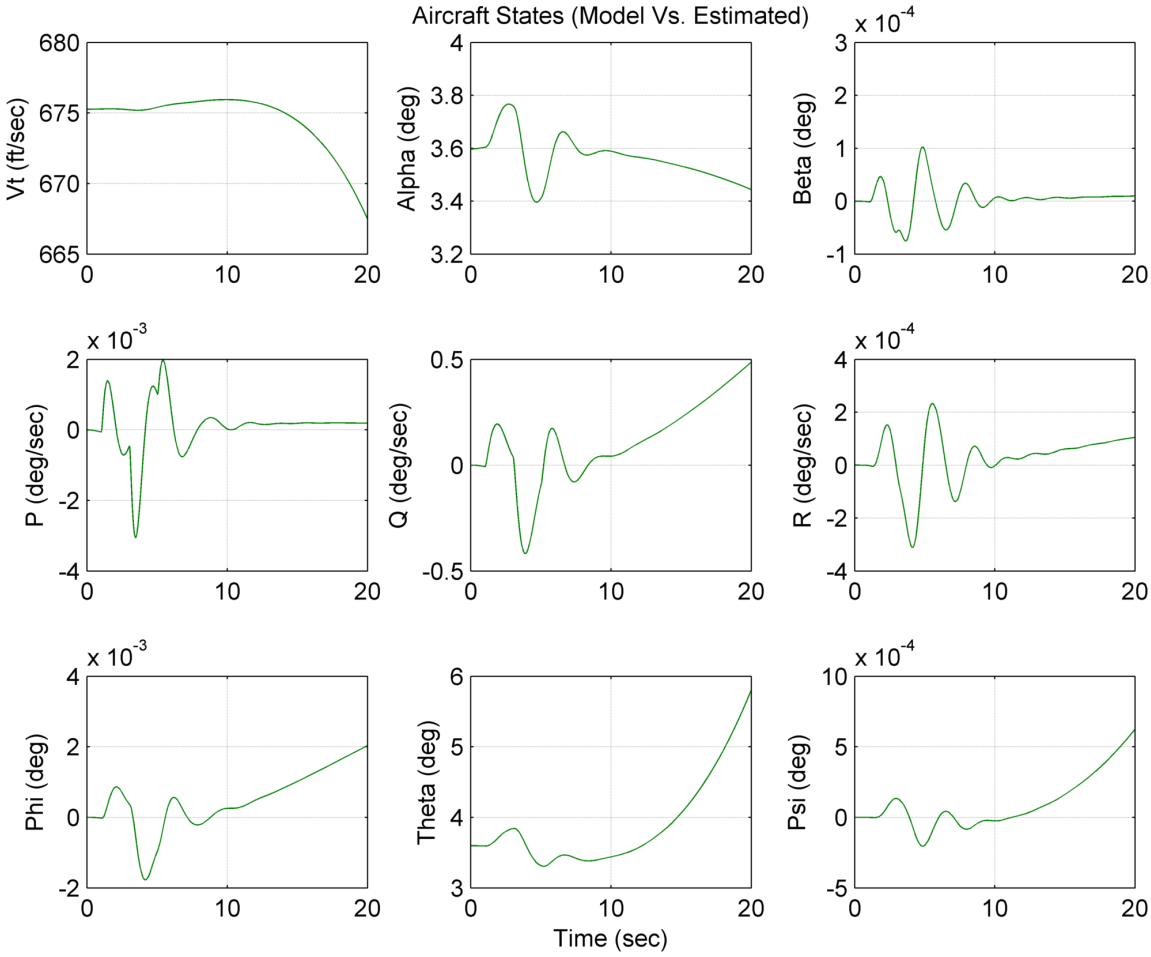

Figure 4.43: $\Delta\hat{C}_{m_q}$ Test Case 3, Aircraft State Tracking

C_L

The aircraft model defined in Section 2.2 and the aircraft pitching moment coefficient, C_L, estimator derived in Section 3.4.1 have been simulated together to verify proper convergence of the estimated parameter ($\Delta\hat{C}_{L_\alpha}$ to the actual parameter (ΔC_{L_α}. This results section contains plots illustrating the convergence of the estimated parameter along with the tracking of the model reference aircraft states to the estimated states for the unknown C_L parameter.

$\Delta\hat{C}_{L_\alpha}$ Results

The change in lift coefficient due to a change in angle of attack, $\Delta\hat{C}_{L_\alpha}$, estimator derived in Section 3.4.1 has been programmed and simulated with the results shown in this section. Table 4.9 shows the three different test conditions and results that are documented in this section, the variables that can be changed are the aircraft control command, the actual value of $\hat{C}_{L_\alpha}$, the upper and lower $\hat{C}_{L_\alpha}$, and η. For all of the $\Delta\hat{C}_{L_\alpha}$ test cases, $\lambda = 1$. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots below. This section contains results from running the $\Delta\hat{C}_{L_\alpha}$ estimator only, multiple estimators have been run at the same time and the results can be found in Section 4.2.1.

Table 4.9: $\hat{C}_{L_\alpha}$ Test Case Table

Test Case	dhs Command	C_{L_α}	$C_{L_{\alpha_{upp}}}$	$C_{L_{\alpha_{low}}}$	η	Final Value
1	Sin, .05deg, 1Hz	-.01	-.008	-.03	.001	$-1.0688e^{-2}$
2	Sin, .05deg, 1Hz	-.01	-.015	-.03	.0008	$-1.0143e^{-2}$
3	Sin, .3deg, 1Hz	-.01	-.015	-.03	.008	$-1.0098e^{-2}$

$\Delta\hat{C}_{L_\alpha}$ Test Case 1

Figure 4.44 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .05 degree sin wave to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

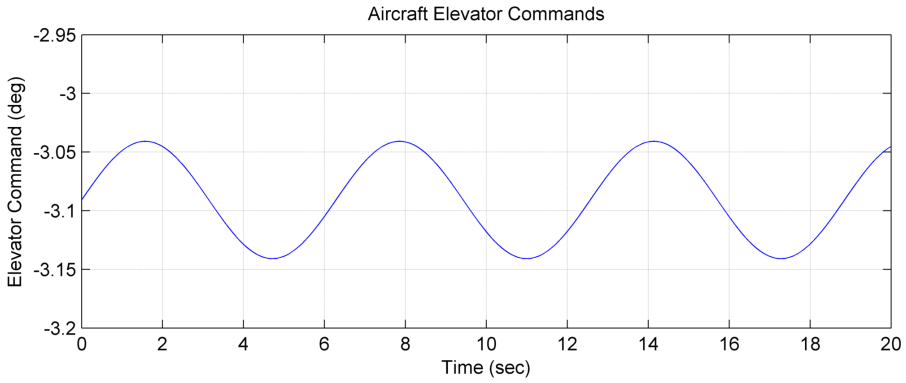

Figure 4.44: $\Delta\hat{C}_{L_\alpha}$ Test Case 1, PID Aircraft Control Commands

Figure 4.45 shows the convergence of the estimated parameter $\Delta\hat{C}_{L_\alpha}$ to the actual parameter $\Delta C_{L_\alpha} = -.01$, the system takes over 20 seconds to converge upon the model reference parameter. This is a significant amount of time compared to all of the other parameters that are estimated. This could be caused by the tuning of the estimator parameters or the definition of the sliding surface. Future work Section 5.2 explains the need for an automated estimator tuning method.

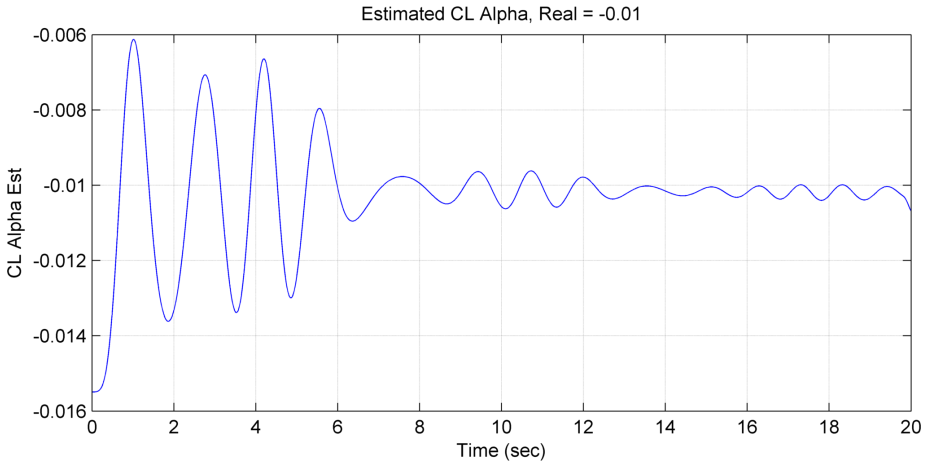

Figure 4.45: $\Delta\hat{C}_{L_\alpha}$ Test Case 1, Converging Upon ΔC_{L_α}

Figure 4.46 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{L_\alpha}$ parameter has converged upon the actual ΔC_{L_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in lift coefficient due to a change in angle of attack, $\Delta\hat{C}_{L_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

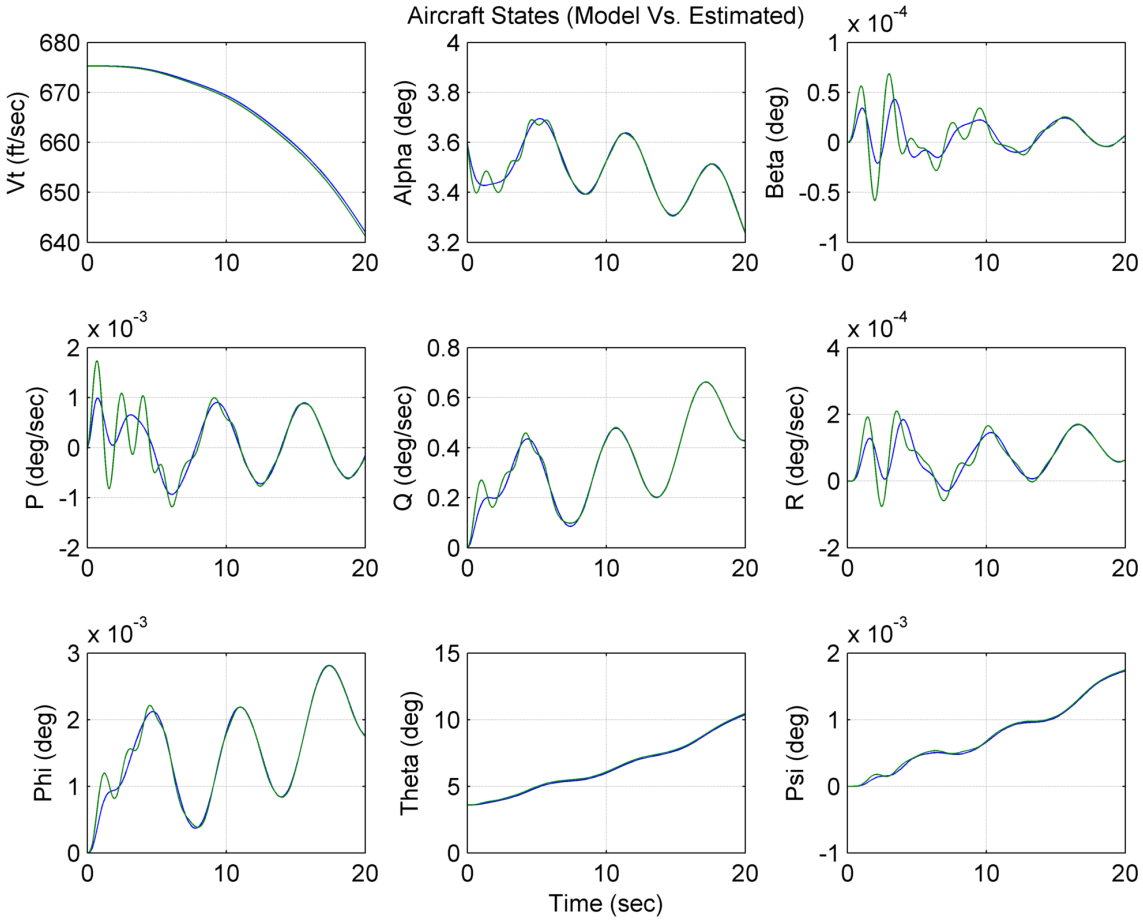

Figure 4.46: $\Delta\hat{C}_{L_\alpha}$ Test Case 1, Aircraft State Tracking

$\Delta\hat{C}_{L_\alpha}$ Test Case 2

Figure 4.47 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .05 degree sin wave to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

Figure 4.47: $\Delta\hat{C}_{L_\alpha}$ Test Case 2, PID Aircraft Control Commands

Figure 4.48 shows the convergence of the estimated parameter $\Delta\hat{C}_{L_\alpha}$ to the actual parameter $\Delta C_{L_\alpha} = -.01$, the system takes over 20 seconds to converge upon the model reference parameter. This is a significant amount of time compared to all of the other parameters that are estimated. This could be caused by the tuning of the estimator parameters or the definition of the sliding surface. Future work Section 5.2 explains the need for an automated estimator tuning method. The starting upper and lower bounds have been changed in this example, as well as the η parameter. This has cause an increase accuracy and decreased the convergence time, however the time to estimation is still higher that desired.

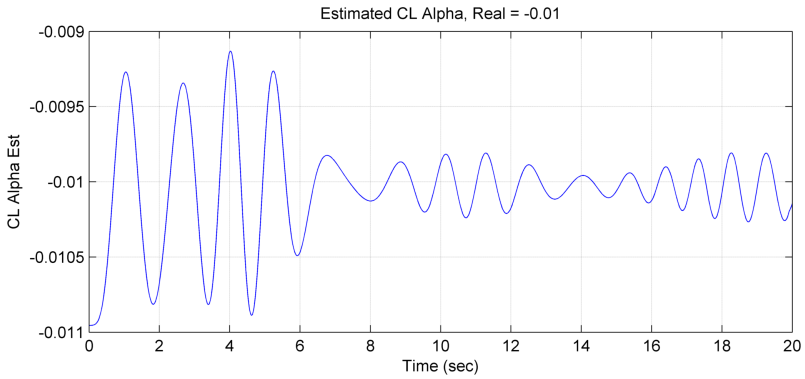

Figure 4.48: $\Delta\hat{C}_{L_\alpha}$ Test Case 2, Converging Upon ΔC_{L_α}

Figure 4.49 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{L_\alpha}$ parameter has converged upon the actual ΔC_{L_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in lift coefficient due to a change in angle of attack, $\Delta\hat{C}_{L_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

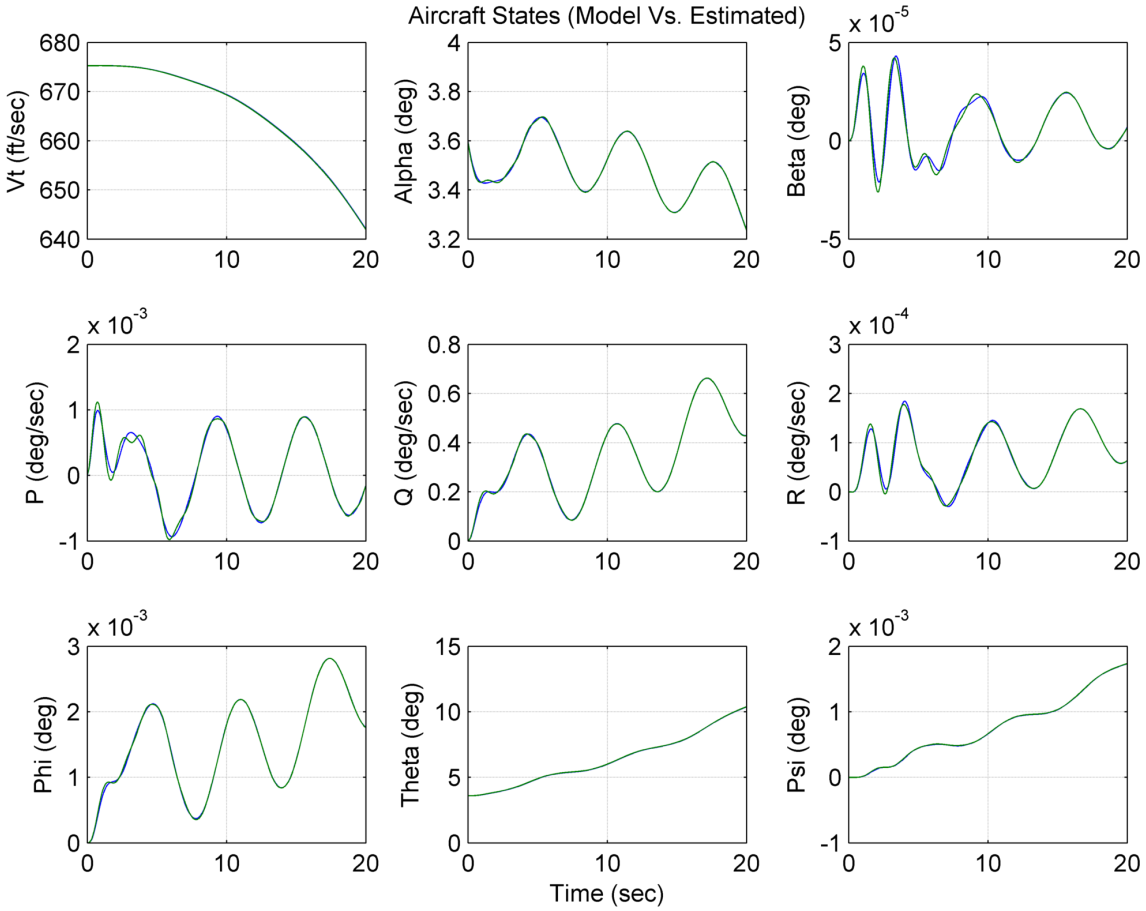

Figure 4.49: $\Delta\hat{C}_{L_\alpha}$ Test Case 2, Aircraft State Tracking

$\Delta\hat{C}_{L_\alpha}$ Test Case 3

Figure 4.50 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .3 degree sin wave to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

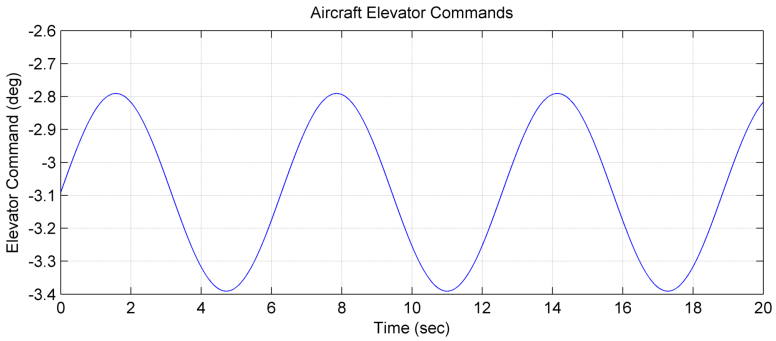

Figure 4.50: $\Delta\hat{C}_{L_\alpha}$ Test Case 3, PID Aircraft Control Commands

Figure 4.51 shows the convergence of the estimated parameter $\Delta\hat{C}_{L_\alpha}$ to the actual parameter $\Delta C_{L_\alpha} = -.01$, the system takes over 20 seconds to converge upon the model reference parameter. This is a significant amount of time compared to all of the other parameters that are estimated. This could be caused by the tuning of the estimator parameters or the definition of the sliding surface. Future work Section 5.2 explains the need for an automated estimator tuning method. The input to the system was increased for this example to try and decrease the amount of time required to converge upon the actual value, the estimator did in-fact converge faster, however it still takes over fifteen seconds to reach a reasonable estimate.

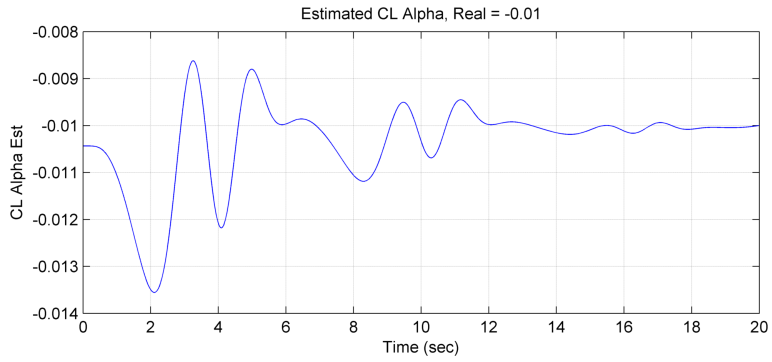

Figure 4.51: $\Delta\hat{C}_{L_\alpha}$ Test Case 3, Converging Upon ΔC_{L_α}

Figure 4.52 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{L_\alpha}$ parameter has converged upon the actual ΔC_{L_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in lift coefficient due to a change in angle of attack, $\Delta\hat{C}_{L_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

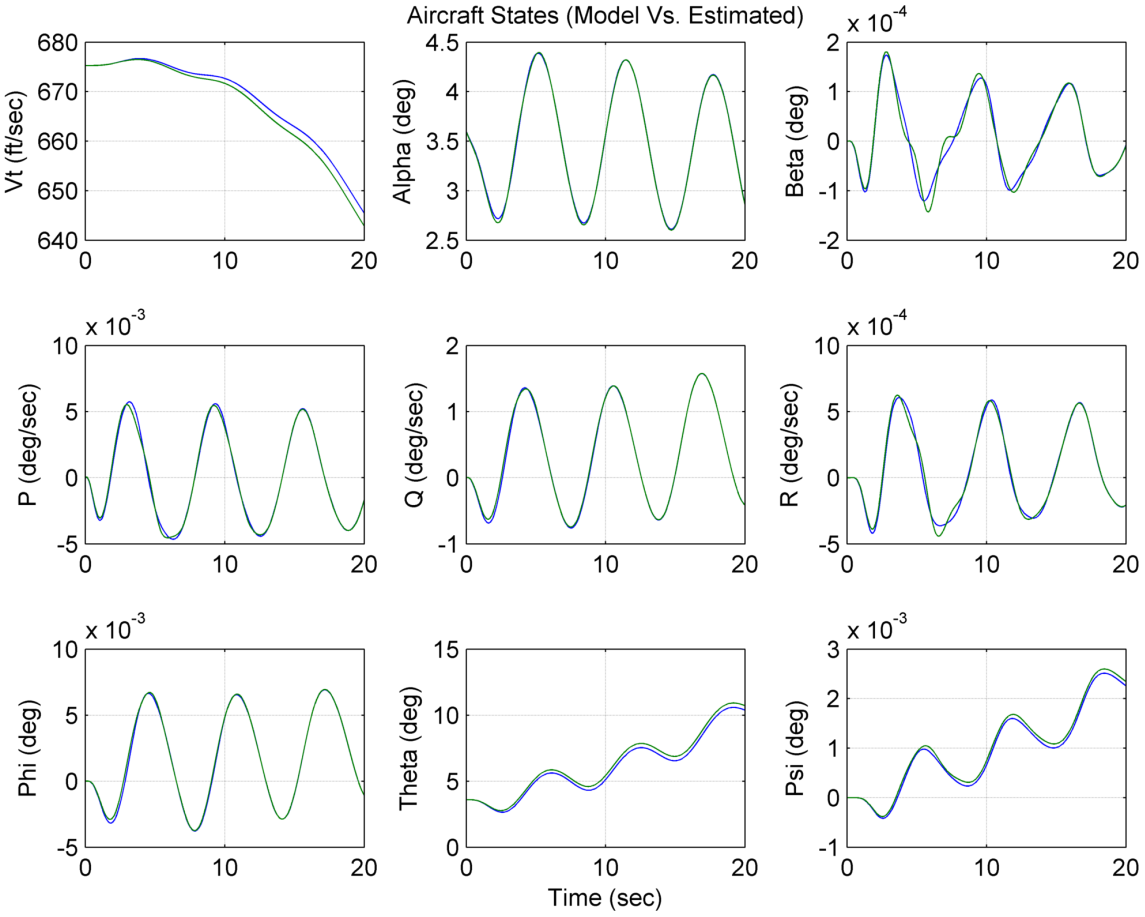

Figure 4.52: $\Delta\hat{C}_{L_\alpha}$ Test Case 3, Aircraft State Tracking

C_D

The aircraft model defined in Section 2.2 and the aircraft drag coefficient, C_D, estimator derived in Section 3.4.1 have been simulated together to verify proper convergence of the estimated parameter ($\Delta\hat{C}_{D_\alpha}$ to the actual parameter (ΔC_{D_α}. This results section contains plots illustrating the convergence of the estimated parameter along with the tracking of the model reference aircraft states to the estimated states for the unknown C_D parameter.

$\Delta\hat{C}_{D_\alpha}$ Results

The change in drag coefficient due to a change in angle of attack, $\Delta\hat{C}_{D_\alpha}$, estimator derived in Section 3.4.1 has been programmed and simulated with the results shown in this section. Table 4.10 shows the two different test conditions and results that are documented in this section, the variables that can be changed are the aircraft control command, the actual value of $\hat{C}_{D_\alpha}$, and the upper and lower $\hat{C}_{D_\alpha}$. For all of the $\Delta\hat{C}_{D_\alpha}$ test cases, $\lambda = 1$ and $\eta = 1000$. These modifications change the convergence characteristics of the estimator, the differences are illustrated in the plots below. This section contains results from running the $\Delta\hat{C}_{D_\alpha}$ estimator only, multiple estimators have been run at the same time and the results can be found in Section 4.2.1.

Table 4.10: $\hat{C}_{D_\alpha}$ Test Case Table

Test Case	dhs Command	C_{L_α}	$C_{L_{\alpha_{upp}}}$	$C_{L_{\alpha_{low}}}$	Final Value
1	Sin, .05deg, 1Hz	.001	.005	.0008	$1.0000e^{-3}$
2	Doublet, -.05deg	.001	.1	$8e^{-8}$	$1.0000e^{-3}$

$\Delta\hat{C}_{D_\alpha}$ Test Case 1

Figure 4.53 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .05 degree sin wave to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

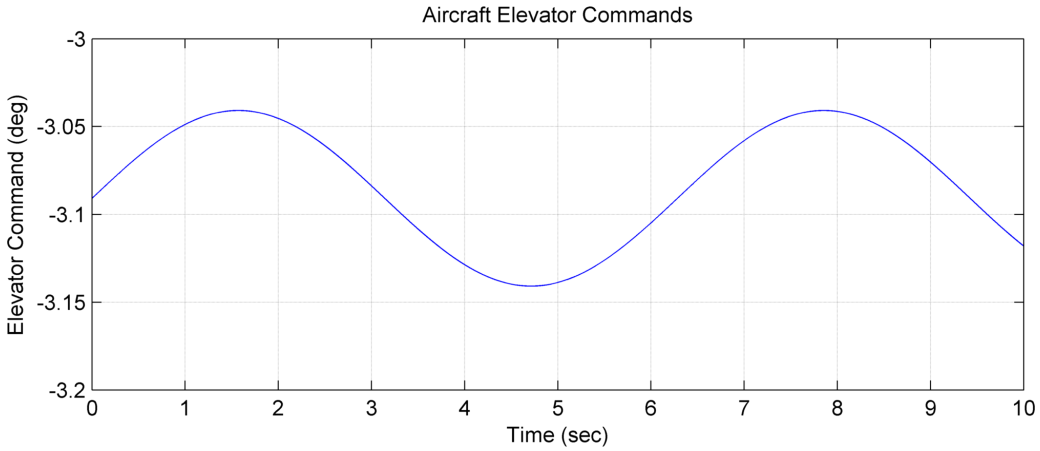

Figure 4.53: $\Delta\hat{C}_{D_\alpha}$ Test Case 1, PID Aircraft Control Commands

Figure 4.54 shows the convergence of the estimated parameter $\Delta\hat{C}_{D_\alpha}$ to the actual parameter $\Delta C_{D_\alpha} = .001$, the system takes about 250 milliseconds to converge upon the model reference parameter. This is a very small amount of time to estimation and a desirable result.

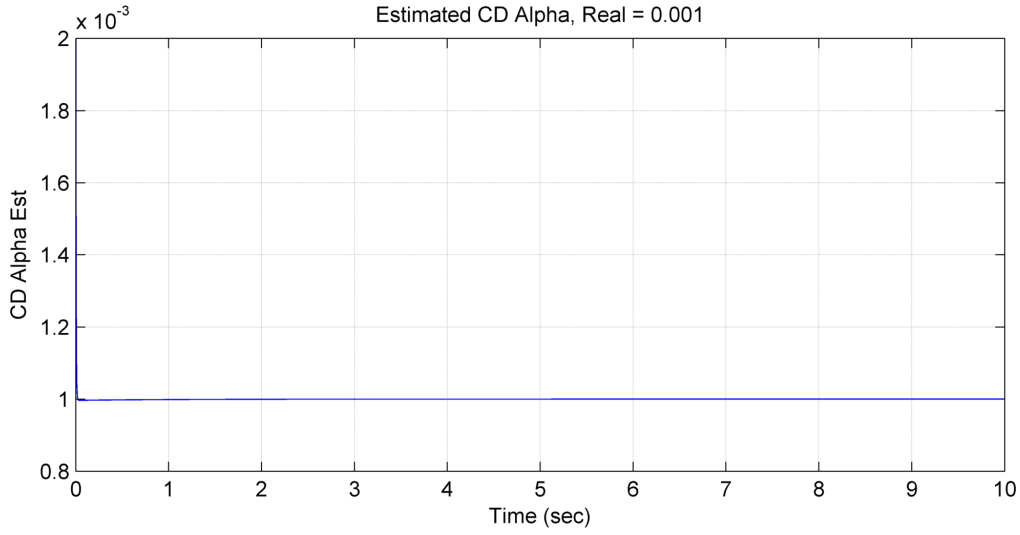

Figure 4.54: $\Delta\hat{C}_{D_\alpha}$ Test Case 1, Converging Upon ΔC_{D_α}

Figure 4.55 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{D_\alpha}$ parameter has converged upon the actual ΔC_{D_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in lift coefficient due to a change in angle of attack, $\Delta\hat{C}_{D_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

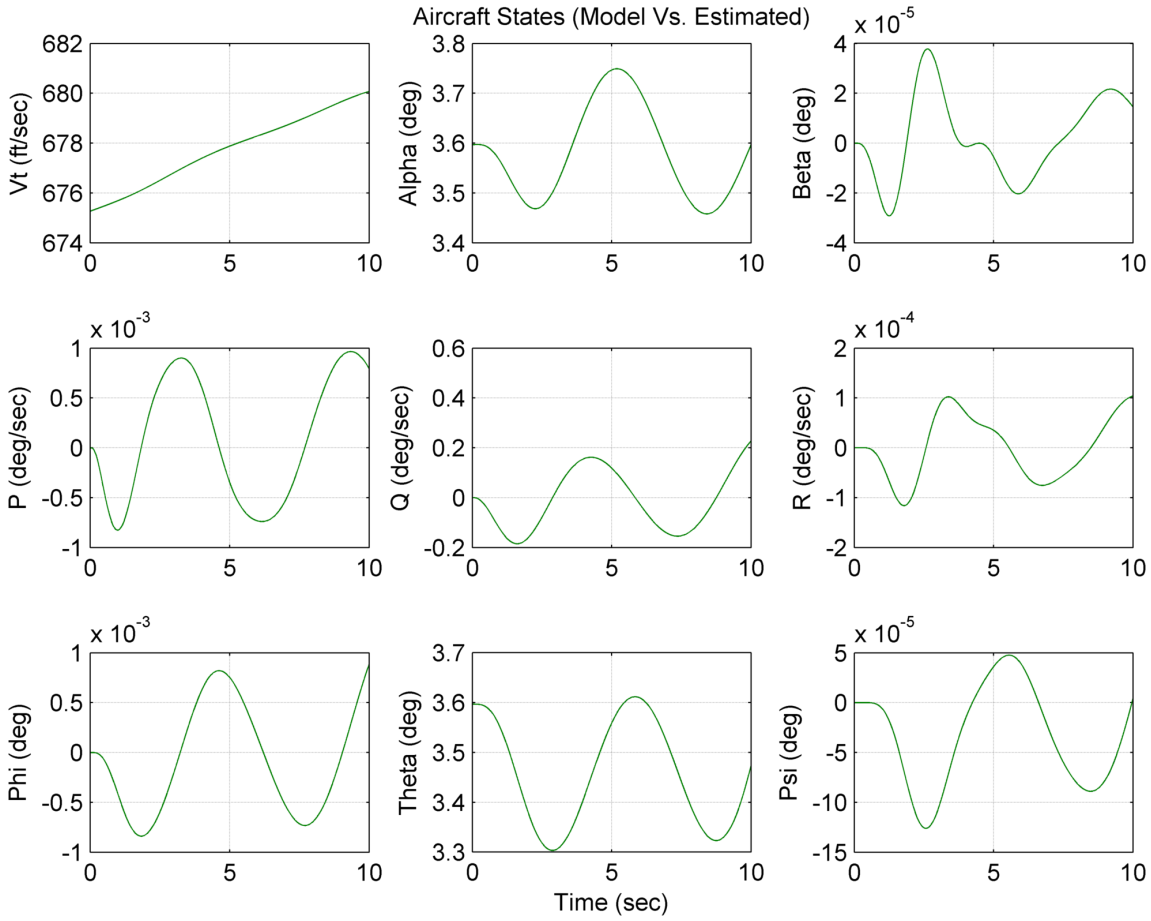

Figure 4.55: $\Delta\hat{C}_{D_\alpha}$ Test Case 1, Aircraft State Tracking

$\Delta\hat{C}_{D_\alpha}$ Test Case 2

Figure 4.56 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .05 degree doublet command to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters.

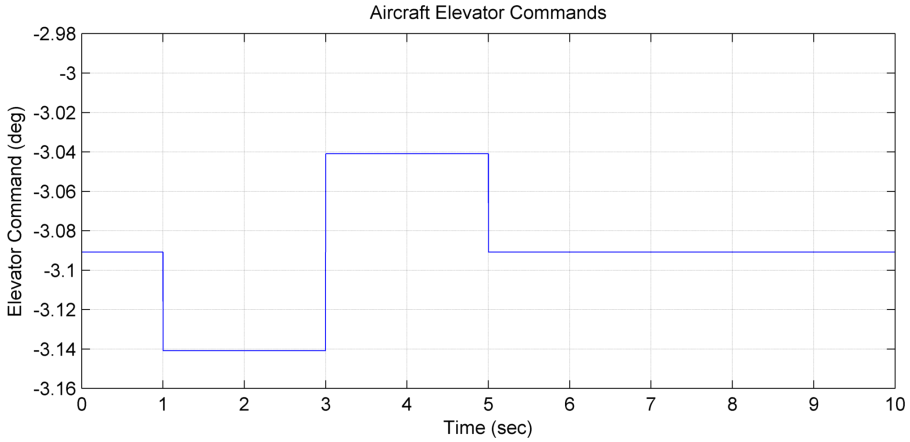

Figure 4.56: $\Delta\hat{C}_{D_\alpha}$ Test Case 2, PID Aircraft Control Commands

Figure 4.57 shows the convergence of the estimated parameter $\Delta\hat{C}_{D_\alpha}$ to the actual parameter $\Delta C_{D_\alpha} = .001$, the system takes about 250 milliseconds to converge upon the model reference parameter. This is a very small amount of time to estimation and a desirable result.

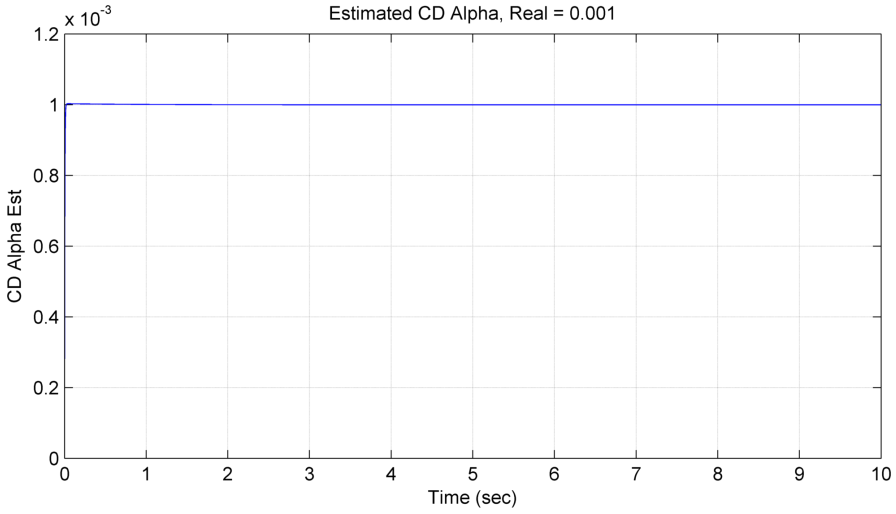

Figure 4.57: $\Delta\hat{C}_{D_\alpha}$ Test Case 2, Converging Upon ΔC_{D_α}

Figure 4.58 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated $\Delta\hat{C}_{D_\alpha}$ parameter has converged upon the actual ΔC_{D_α} the aircraft states are tracked perfectly. In this test the aerodynamic coefficient estimator for the change in lift coefficient due to a change in angle of attack, $\Delta\hat{C}_{D_\alpha}$, successfully estimates the correct parameter and the system states are tracked as expected.

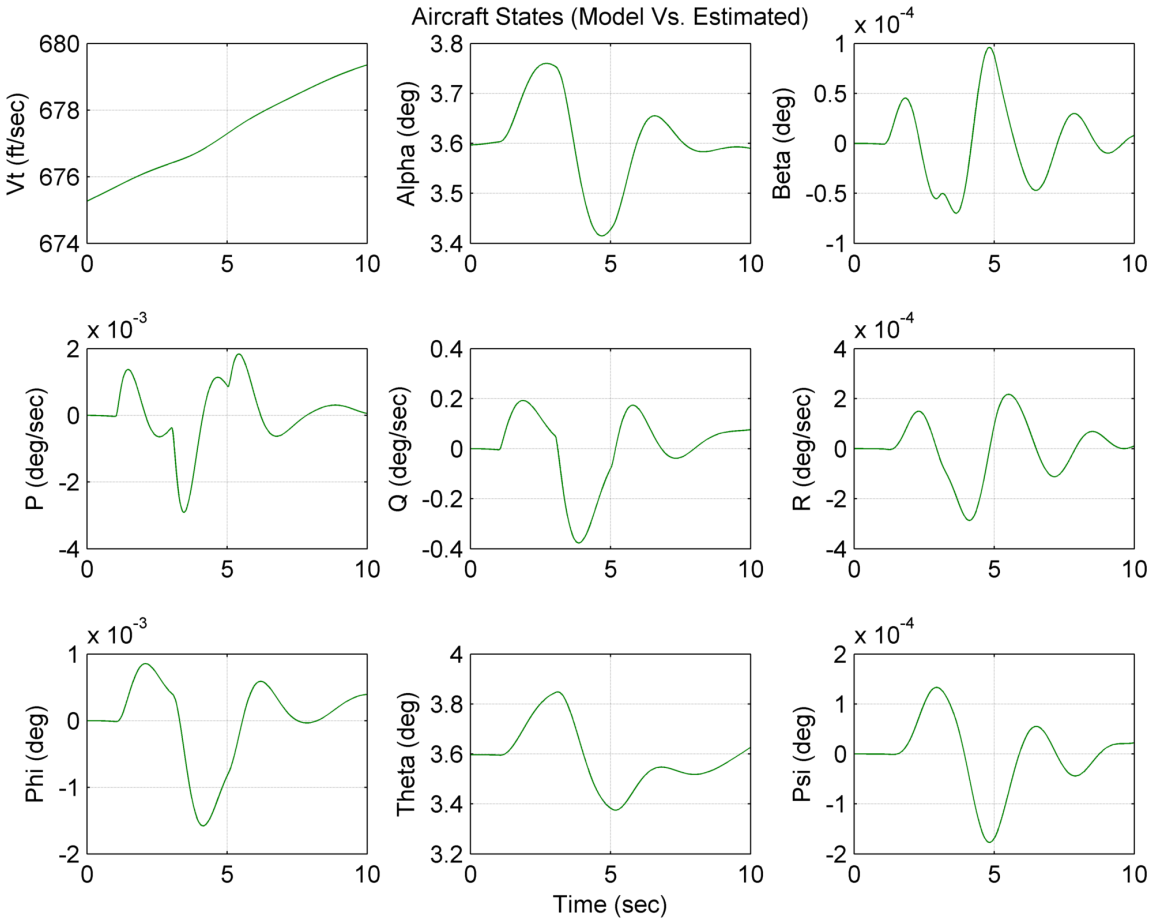

Figure 4.58: $\Delta\hat{C}_{D_\alpha}$ Test Case 2, Aircraft State Tracking

Longitudinal Results Summary

There have been four estimators developed for the longitudinal aircraft model for each of the following aerodynamic parameters, $\Delta\hat{C}_{m_\alpha}$, $\Delta\hat{C}_{m_q}$, $\Delta\hat{C}_{L_\alpha}$, and $\Delta\hat{C}_{D_\alpha}$, these four estimators were analyzed in detail and the results of each individual estimator performance is shown in Section 4.2.1. Once each of the estimators performance was verified independently of the others, then they were combined into a single simulation with a doublet input to the elevator of $-.5$ degrees, the simulation results are shown in this section.

Table 4.11 displays the longitudinal model parameter identification results for the elevator doublet maneuver. Simulation runs were conducted obtaining time history data with the longitudinal increment factor shown by the truth column in Table 4.11. The identified aerodynamic increment factors shown in Table 4.11 are very close to the truth model after a simulation time of 20 seconds. The estimator convergence parameters can be optimized to achieve better convergence performance, this type of estimator optimization is discussed in Section 5.

Table 4.11: Combined Parameter Estimation Results Table

Parameters	Truth	Estimated
$\Delta\hat{C}_{m_\alpha}$ (1/deg)	-0.0011	$-1.0989e^{-3}$
$\Delta\hat{C}_{m_q}$ (1/deg/s)	.01	$9.9997e^{-3}$
$\Delta\hat{C}_{L_\alpha}$ (1/deg)	-.01	$-1.0019e^{-2}$
$\Delta\hat{C}_{D_\alpha}$ (1/deg)	.001	$9.9998e^{-4}$

Figure 4.59 shows the inputs to the aircraft system for this test case, the only input that is used in this case is a .5 degree doublet command to the elevator. This command input will excite the longitudinal model and allows for the estimation of all four of the longitudinal parameters.

Figure 4.59: Combined PID Elevator Input

Figure 4.60 shows the convergence of all four estimated parameters, each of the estimated parameters take a different amount of time to fully estimate. This difference is estimation is caused by the tuning of the estimation parameters. Using a more optimized tuning function these results would look similar. Some of the aircraft coefficients are easier to estimate than others, this is shown true here as the $\Delta\hat{C}_{D_\alpha}$ is estimated almost instantly, while the others continue to converge.

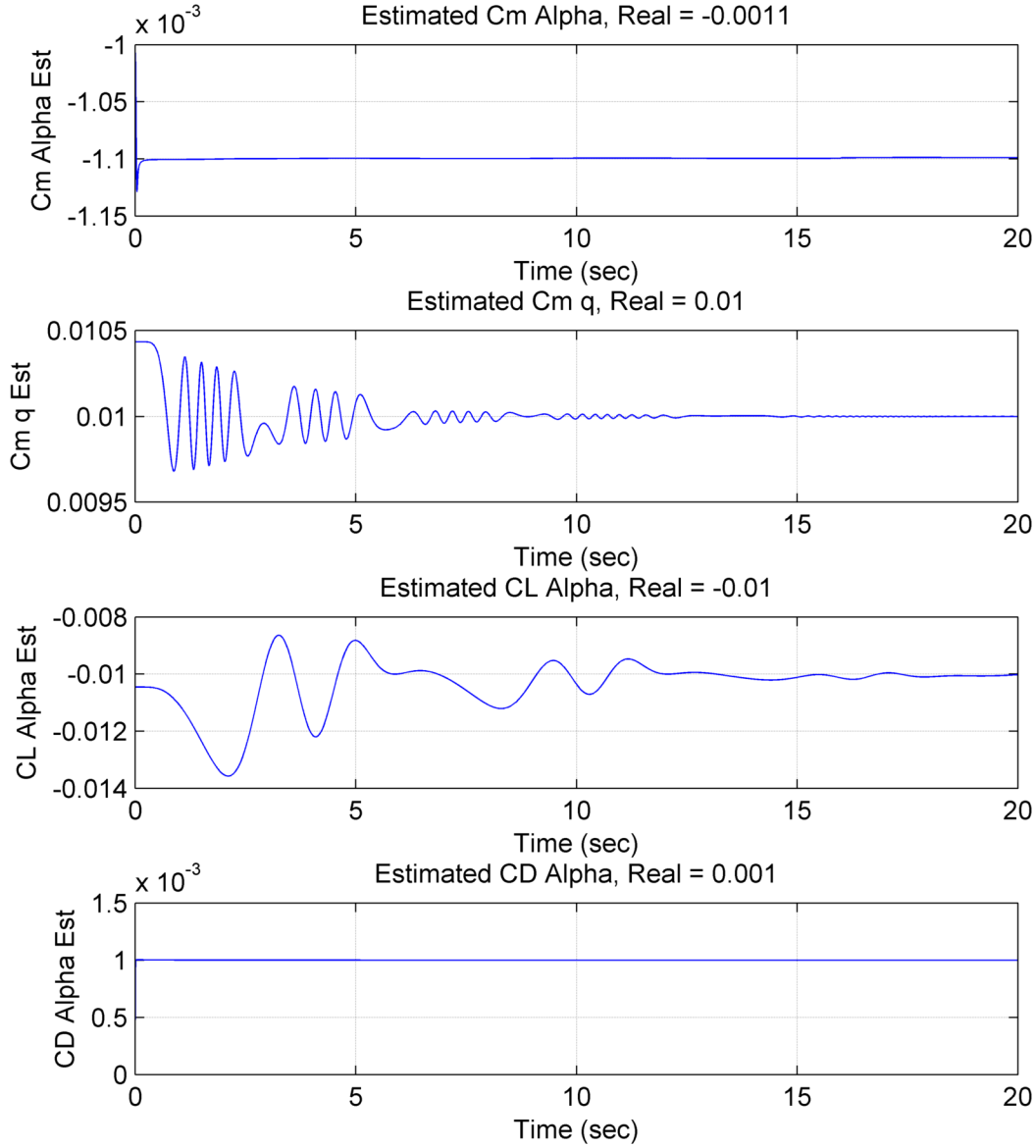

Figure 4.60: Combined PID Parameter Convergence

Figure 4.61 shows the model reference aircraft states in blue and the estimated aircraft states in green. After the estimated parameters have converged upon the actual system parameters the aircraft states are tracked perfectly. This showcases the ability of the estimators to identify coefficients that allow for the successful tracking of the aircraft states when all four parameters are estimated.

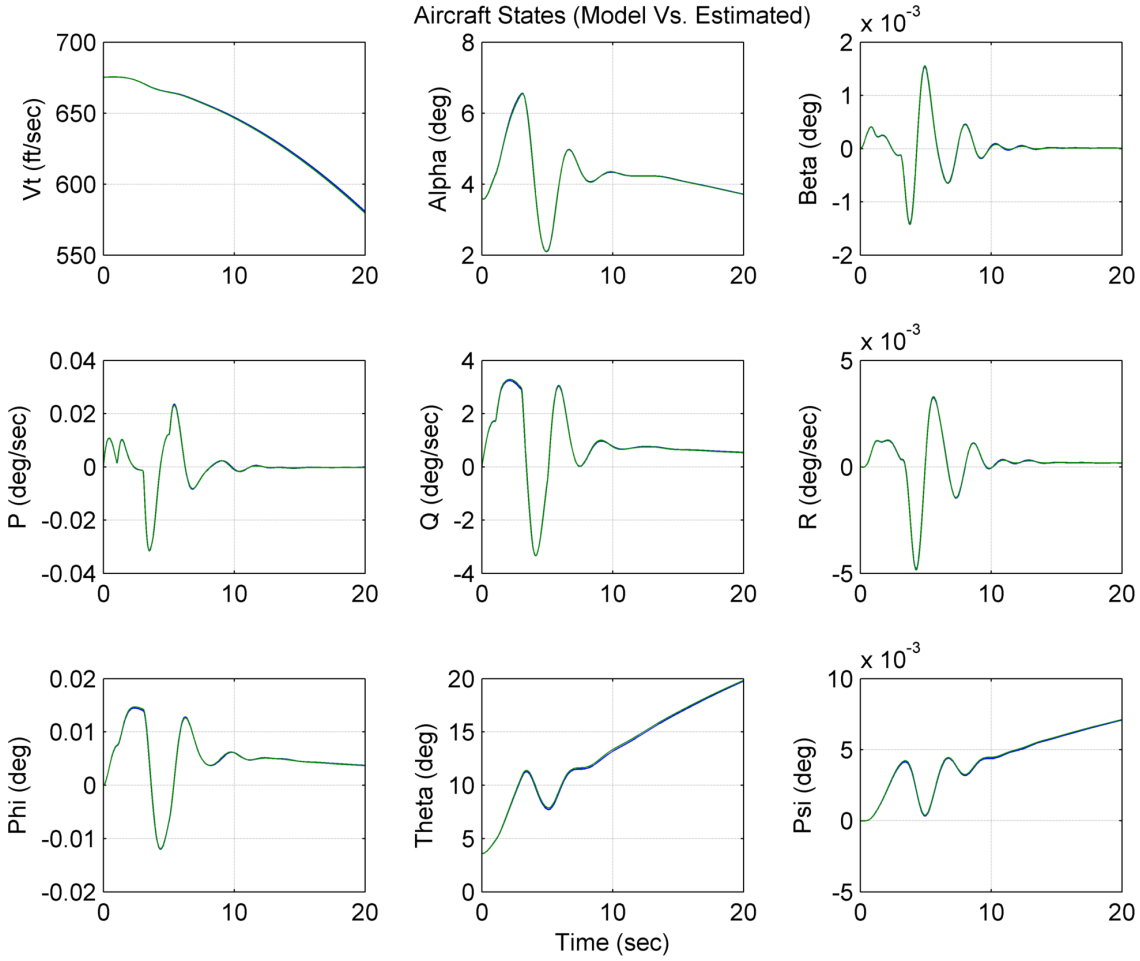

Figure 4.61: Combined PID Aircraft State Tracking

The estimators are successfully able to estimate each of the four coefficients simultaneously for a given doublet maneuver to the elevator, this proves the system is fully capable of estimation and a discussion of results and future work can be found in Section 5.

4.2.2 Comparison with Model Reference Estimator

The model reference estimator that has been developed in Section 3.1.1 is simulated in Matlab and Simulink to estimate the $\Delta\hat{C}_{m_q}$ parameter of the aircraft model. The convergence of the model reference estimation is compared to the convergence of the newly developed sliding estimator in this section for two different scenarios. The first scenario is the estimation of $\Delta\hat{C}_{m_q} = .01$ and the second scenario is the estimation of $\Delta\hat{C}_{m_q} = -.01$.

Scenario 1, $\Delta\hat{C}_{m_q} = .01$

Figure 4.62 shows the inputs to the aircraft system for this scenario, the only input that is used in this case is a .05 degree sin command to the elevator. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters. The same input is used for the simulation of both the model reference estimator and the sliding mode parameter estimator.

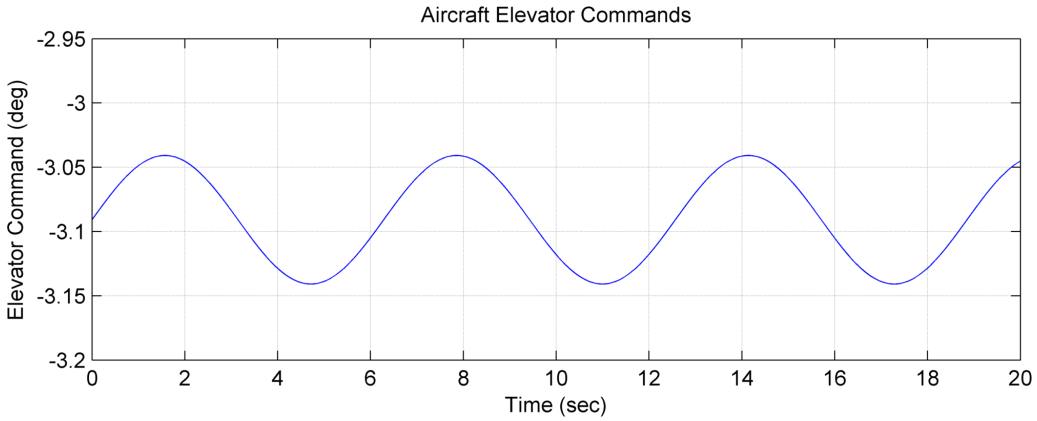

Figure 4.62: Model Reference Scenario 1, PID Aircraft Control Commands

Figure 4.63 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_q}$ to the actual parameter $\Delta\hat{C}_{m_q} = .01$ for both estimators. The sliding estimator is shown as the blue line on the plot and the model reference estimator is shown as the red line. Both estimators converge upon the actual parameter, the convergence rate and accuracy can be changed and can be increased with better tuning of the estimation parameters. For this example the objective is to showcase the ability of both estimators to identify the proper aerodynamic coefficient.

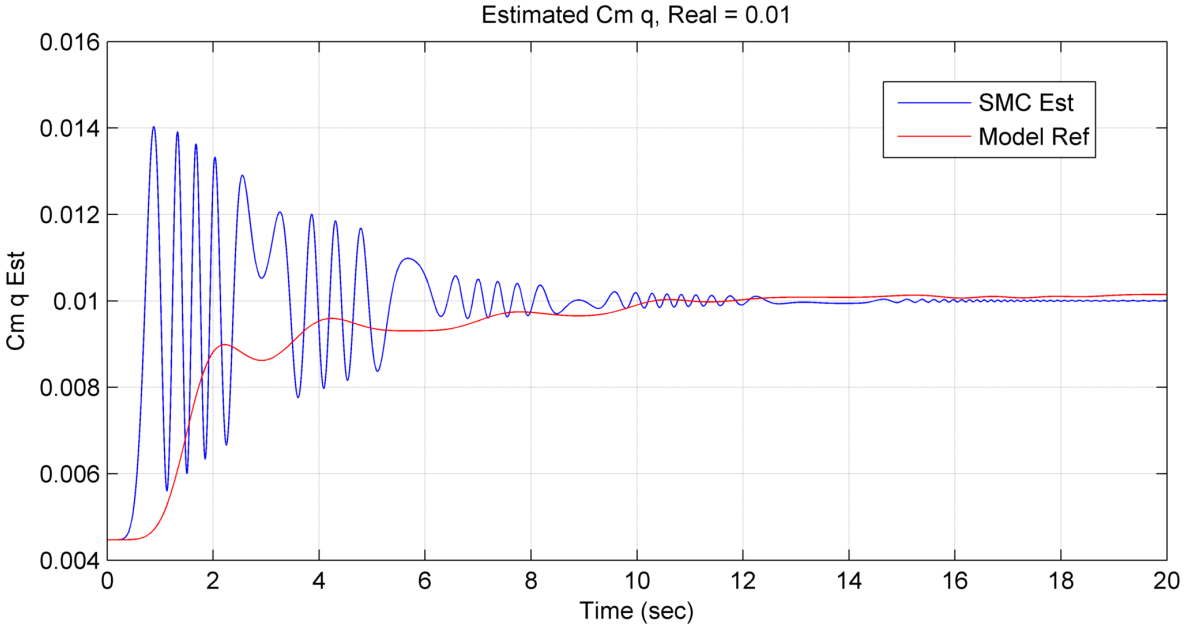

Figure 4.63: Scenario 1, Model Reference and SMC Estimator Convergence Comparison

Scenario 2, $\Delta\hat{C}_{m_q} = -.01$

Figure 4.62 shows the inputs to the aircraft system for this scenario, the only input that is used in this case is a .05 degree sin command to the elevator and this is the same system input as in scenario 1. This command input will excite the longitudinal model and allows for the estimation of longitudinal parameters. The same input is used for the simulation of both the model reference estimator and the sliding mode parameter estimator.

Figure 4.64 shows the convergence of the estimated parameter $\Delta\hat{C}_{m_q}$ to the actual parameter $\Delta\hat{C}_{m_q} = .01$ for the sliding mode estimator only. This scenarios showcases one of the main disadvantages of the model reference estimator, which is the inability to estimate a negative parameter.

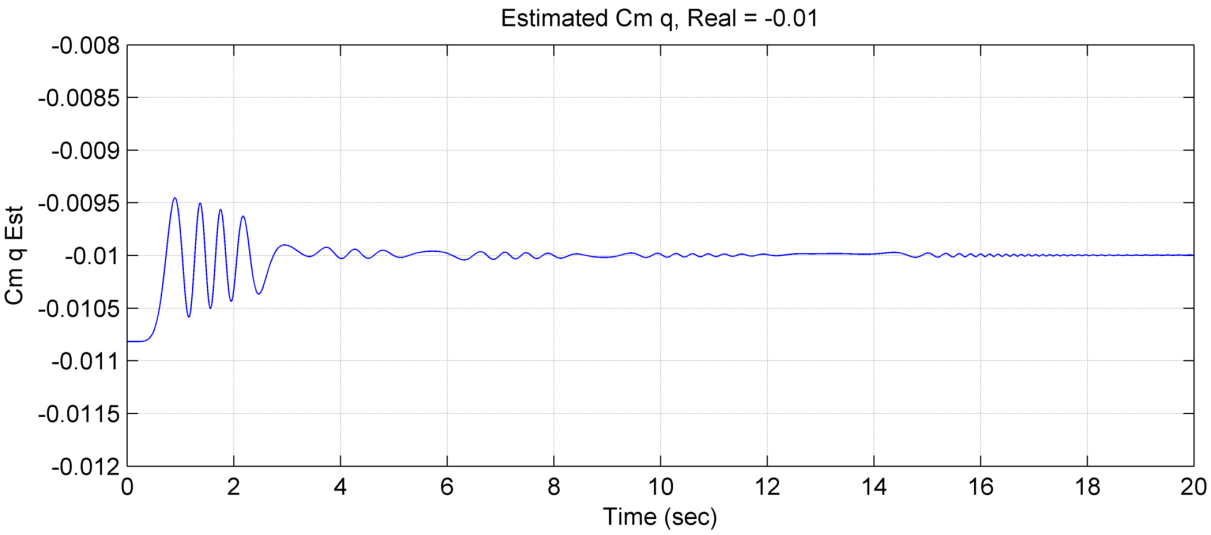

Figure 4.64: Scenario 2, SMC Estimator Convergence

Chapter 5

Conclusion & Future Work

5.1 Conclusion

This work led to the development of a real-time sliding mode parameter estimator for implementation on the aerodynamic coefficients of a nonlinear model. Techniques developed within this work were compared to the previously used methods for parameter estimation, specifically the model reference parameter estimator. It was shown that the newly developed estimator provides comparable performance to previously developed estimators. The newly developed estimator is superior to the model reference estimator because it can guarantee Lyapunov stability and convergence in addition to the ability to estimate negative parameters. The results of this work are beneficial as the need for adaptive control systems increases in both military and commercial aircraft applications.

The newly developed parameter estimation process was implemented on a Lockheed-Martin F-16 aircraft. Results from the nonlinear simulations showed that the estimator was able to successfully estimate four of the aerodynamic coefficients with a high degree of accuracy and fast convergence time. In addition to the running the estimators individually, the estimators were run together and still were able to correctly identify the aerodynamic coefficients.

The end result of this work are methods for parameter estimation shown to work for first order, second order, and nonlinear aircraft coefficients. The estimator development process has been shown to guarantee stability and convergence to the unknown parameter.

5.2 Future Work

The product of this work is a process for the development of a parameter estimator that can guarantee stability and convergence. The simulation completed demonstrate that the estimators developed can be applied to first and second order systems along with nonlinear aircraft parameters. Future related work can be focused in three main areas, the first area is the identification of additional aerodynamic parameters in both the lateral-directional model and in the longitudinal model, the second is the creation of an automatic optimal estimator gain finder, and the third is the use of parameter identification to detect and predict faults within a system.

There are still a number of aerodynamic coefficients that still need to be estimated, only a subset of the complete set of coefficients was investigated in this work. The entire lateral-directional model needs to have estimators build for each of the parameters and analysis of the convergence capability with aileron and rudder inputs. Additional parameters in the longitudinal model will also need to be estimated and analyzed.

The development of a system to automatically identify optimal gains for the estimators is essential for the estimation of multiple parameters inside of a complex system. Estimating all of the aerodynamic gains in the nonlinear aircraft model would require the tuning of many estimators which could lead to a very extensive set up time. An automated process that would identify estimator gains that give the best estimation performance for a given system should be in place to reduce the amount of time needed for estimator set up and implementation.

The ability of the estimator to identify aerodynamic parameters in real-time gives insight into the current operation of the aircraft. Investigation into how the parameters of the aircraft change when a failure is present along with the ability to identify changes in aerodynamic parameters developed in this work will lead to the ability of the estimator to be used for detection and possibly prediction of aircraft failures.

Bibliography

[1] Timothy R. Moes, Mark S. Smith, and Eugene A. Morelli. Flight investigation of prescribed simultaneous independent surface excitations for real-time parameter identification. NASA/TM-2003-212029, available at `http://www.dfrc.nasa.gov/DTRS/`.

[2] Jacob J. Hageman, Mark S. Smith, and Susan Stachowiak. Integration of online parameter identification and neural network for in-flight adaptive control. NASA/TM-2003-212028, available at `http://www.dfrc.nasa.gov/DTRS/`.

[3] Timothy R. Moes, Gregory K. Noffz, and Kenneth W. Iliff. Results from F-18B stability and control parameter estimation flight tests at high dynamic pressures. NASA/TP-2000-209033, available at `http://www.dfrc.nasa.gov/DTRS/`.

[4] Ibrahim Haskara. Sliding mode estimation and optimization methods in nonlinear control problems. *The Ohio State University*, 1999.

[5] K.Hakiki, B. Mazari, A. Liazid, and S. Djaber. Fault reconstruction using sliding mode observers. *American Journal of Applied Sciences*, 3:1669–1674, 2006.

[6] Roy A. McCann and Mohammad S. Islam. Application of a sliding-mode observer for position and speed estimation in switched reluctance motor drives. *IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS*, 37:51–58, 2001.

[7] M.Rolink, T. Boukhobza, and D.Sauter. High order sliding mode observer for fault actuator estimation and it's application to the three tasks benchmark. PDF available at `hal.archives-ouvertes.fr/docs/00/12/10/29/PDF/mathias.pdf`.

[8] Abdelkrim Benchaib, Ahmed Rachid, Eric Audrezet, and Mohamed Tadjine. Real-time sliding-mode observer and control of an induction motor. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, 46:128–138, 1999.

[9] Kanungo Barada Mohanty. Sensorless sliding mode control of induction motor drives. *IEEE*.

[10] Wen Yu. Stability analysis of visual servoing with sliding-mode estimation and neural compensation. *International Journal of Control, Automation, and Systems*, 4:545–558, 2006.

[11] John Kaneshige and Karen Gundy-Burlet. Integrated neural flight and propulsion control system. *American Institute of Aeronautics and Astronautics*, 2001.

[12] Amuliu Bogdan Proca and John M. Miller. Sensorless sliding-mode control of induction motors using operating condition dependent models. *IEEE TRANSACTIONS ON ENERGY CONVERSION*, 18:205–212, 2003.

[13] Robert F. Stengel. Toward intelligent flight control. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 23:1699–1717, 1993.

[14] A. Mokhtari, A Benallegue, , and B Daachi. Robust inner outer controller and sliding mode observer for a quadrotor uav. *ICGST - ARAS*, 6, 2007.

[15] Karen Gundy-Burlet, K. Krishnakumar, Greg Limes, and Don Bryant. Augmentation of an intelligent flight control system for a simulated c-17 aircraft. *American Institute of Aeronautics and Astronautics*.

[16] R.C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, 1998.

[17] S. Park and C. Kee. Enhanced method for single-antenna gps-based attitude determination. *Aircraft Engineering and Aerospace Technology*, 78:236–243, 2006.

[18] J.J. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[19] M. Ertugrul, 0. Kaynak, and A. Sabanovic. A comparison of various VSS techniques on the control of automated guided vehicles. *Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE*, pages 837–842, 1995.

Appendix A

Simulink Diagrams

This Appendix contains images of the Simulink models used to create the results shown in Section 4.

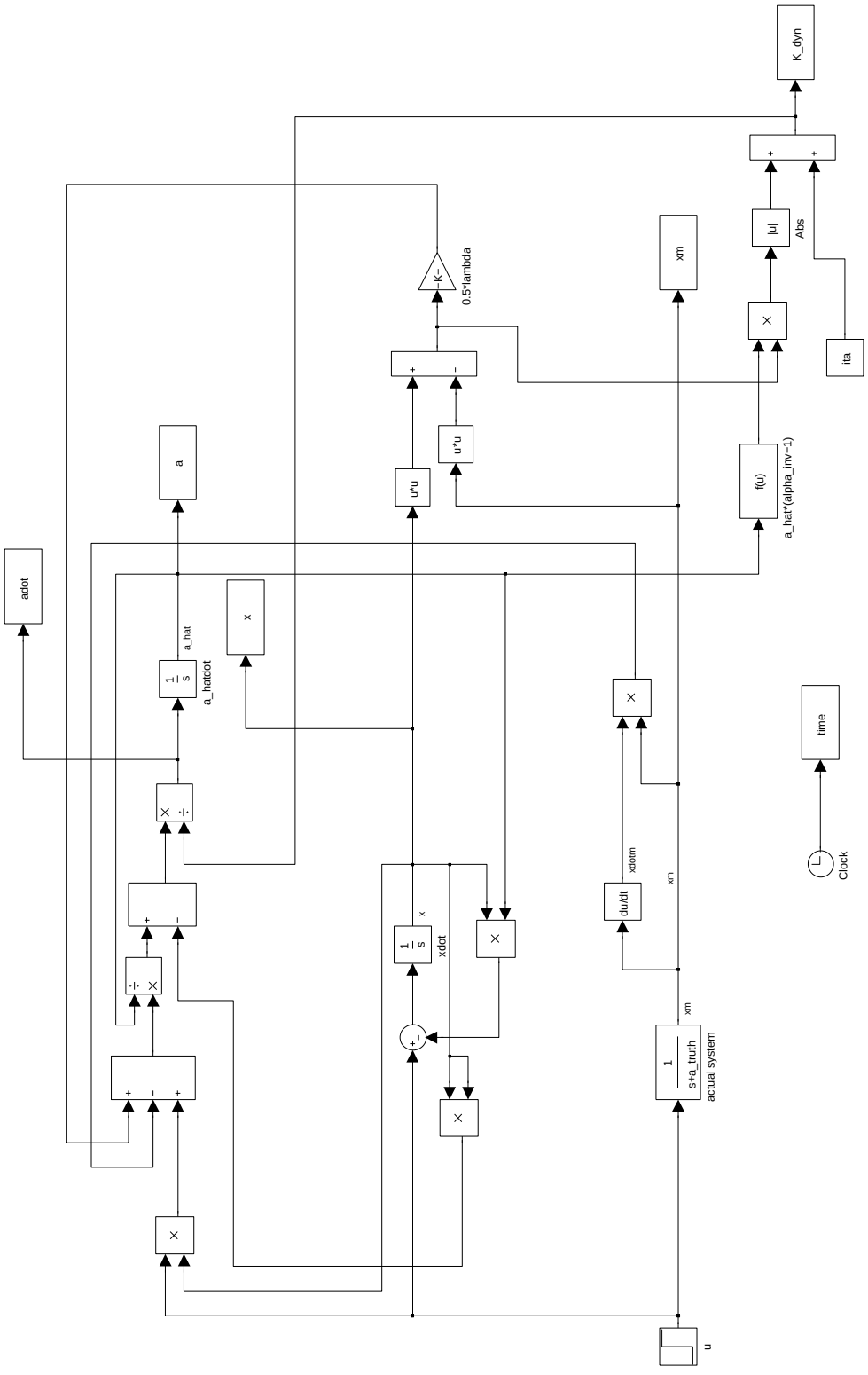

Figure A.1: First Order Model Simulation Diagram

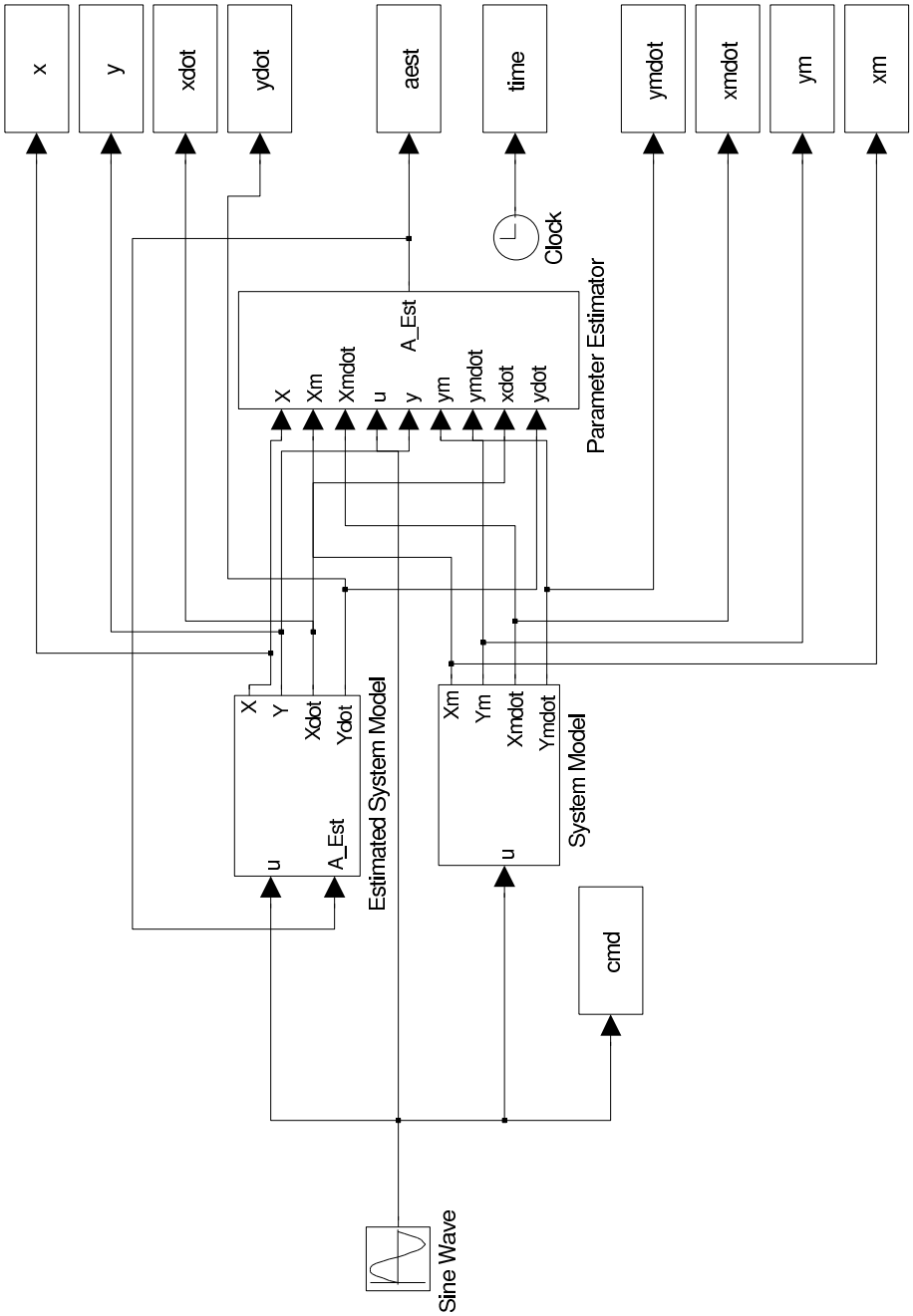

Figure A.2: First Order Nonlinear Model Simulation Overview

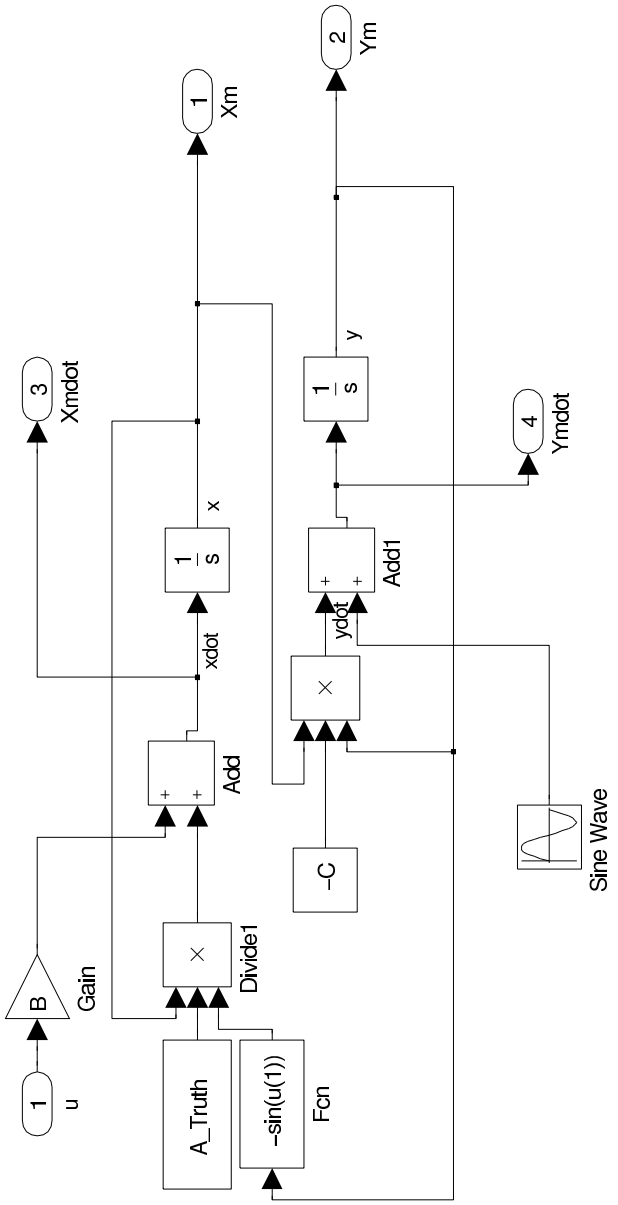

Figure A.3: First Order Nonlinear System Model

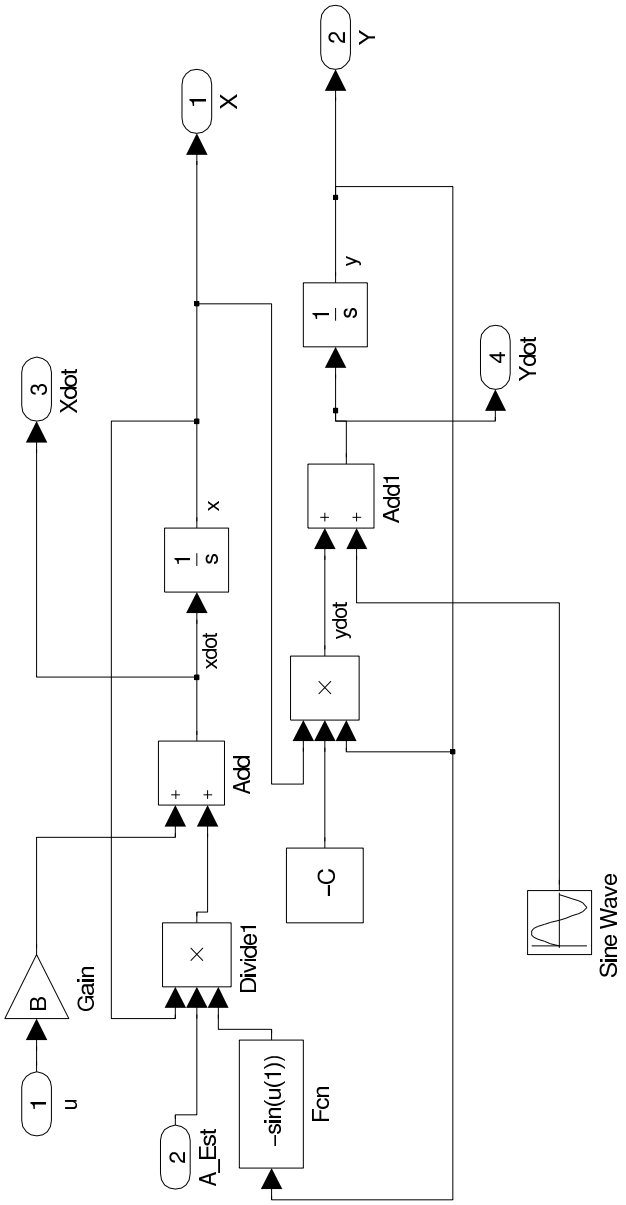

Figure A.4: First Order Nonlinear Estimated System Model

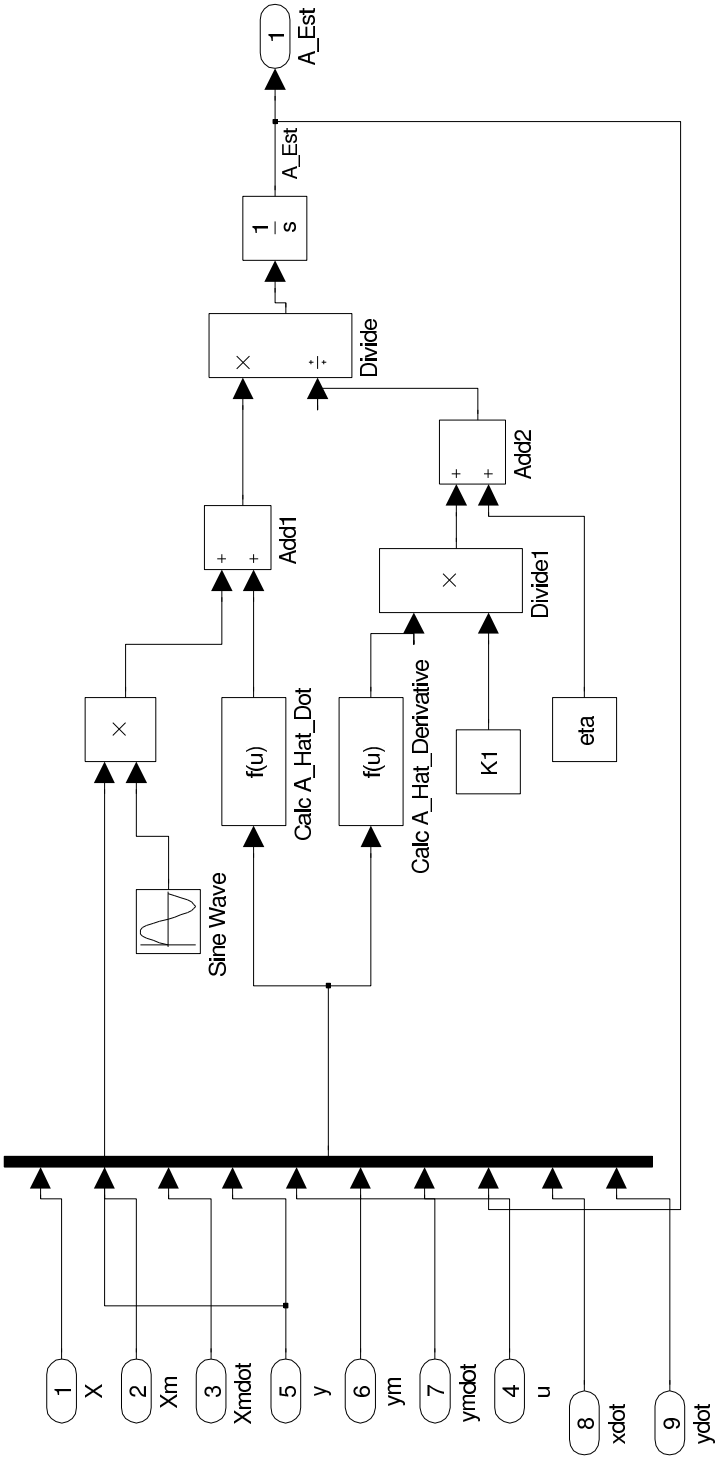

Figure A.5: First Order Nonlinear Model Parameter Estimation Block

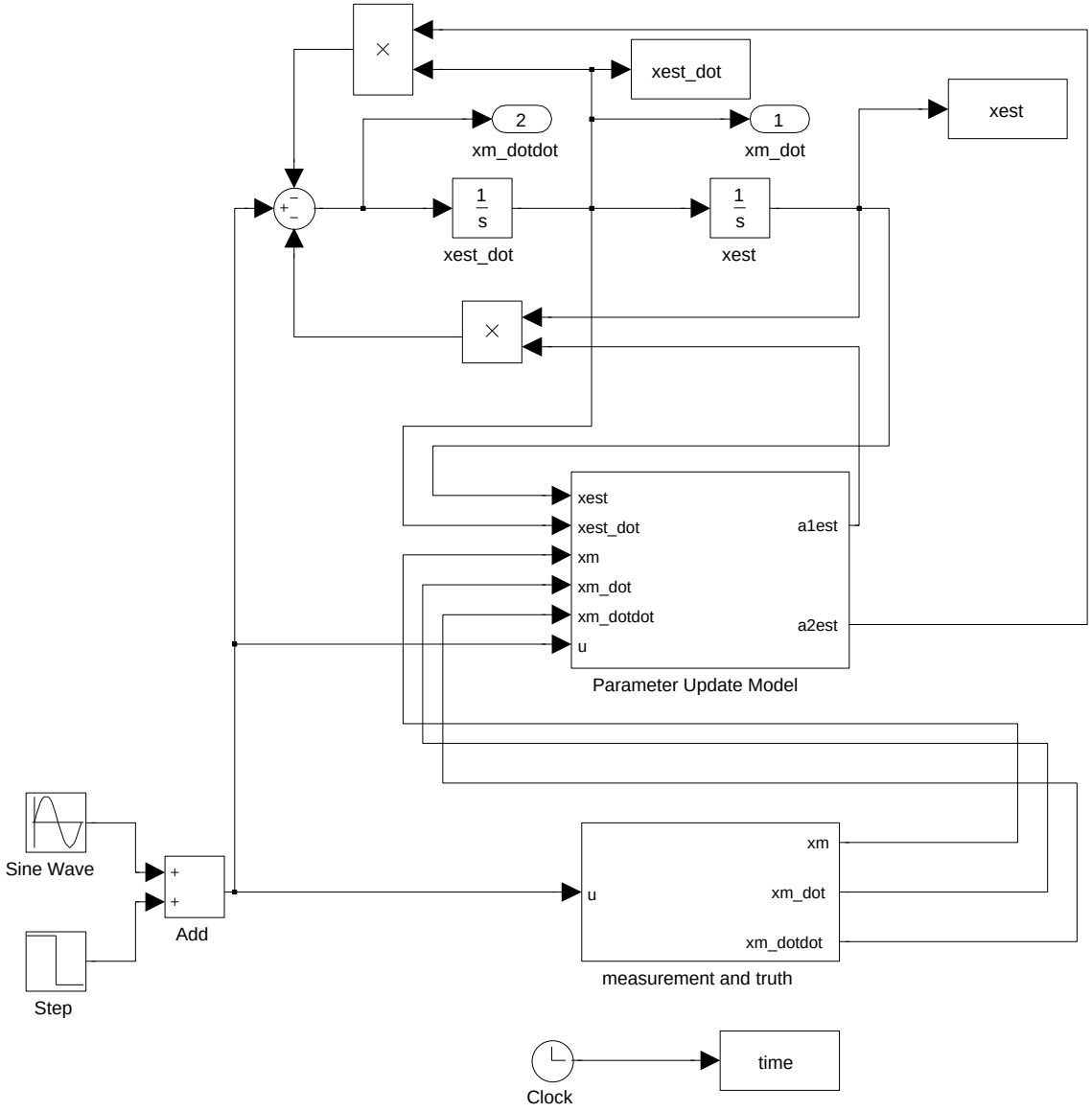

Figure A.6: Second Order Model Simulation Overview

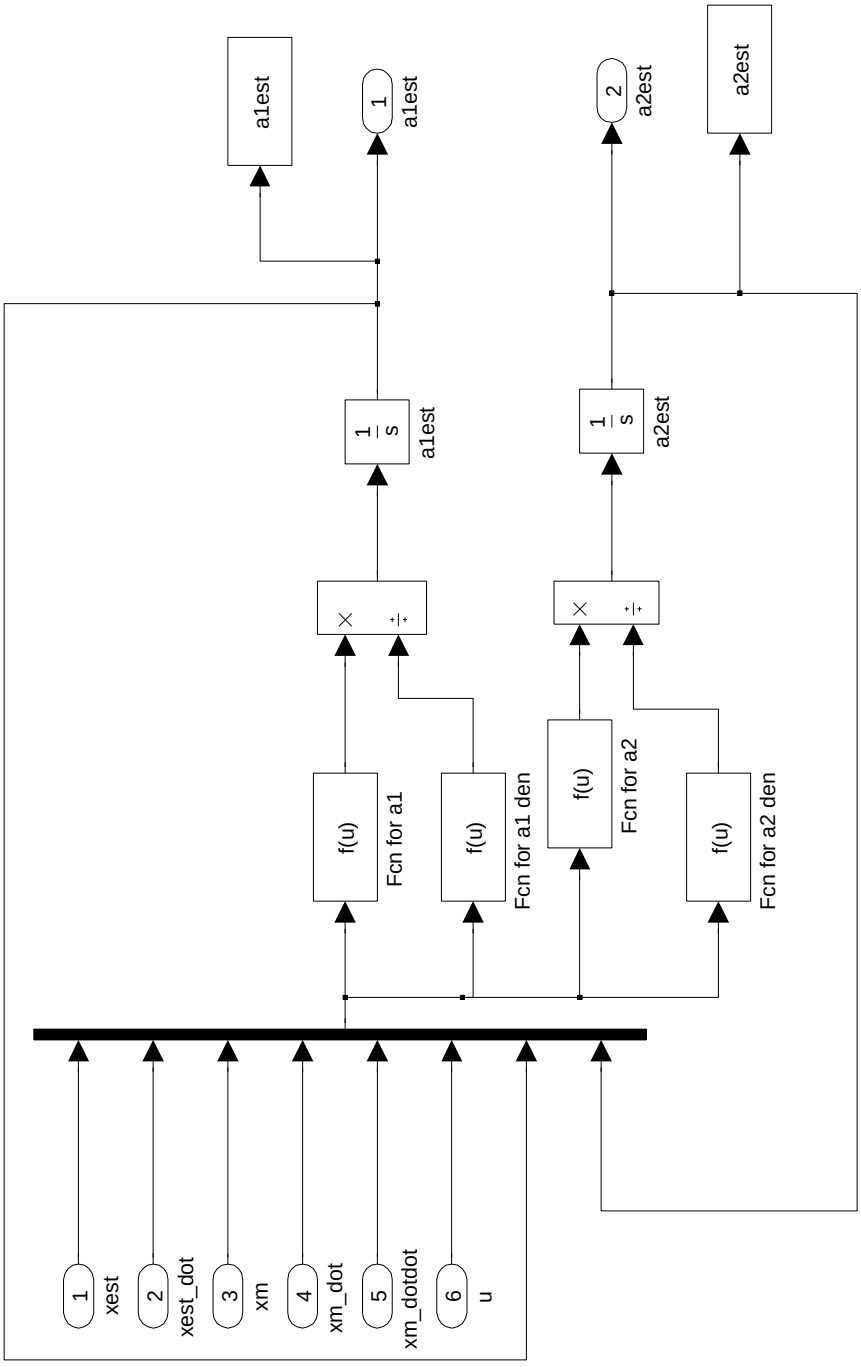

Figure A.7: Second Order Model Parameter Estimation Block

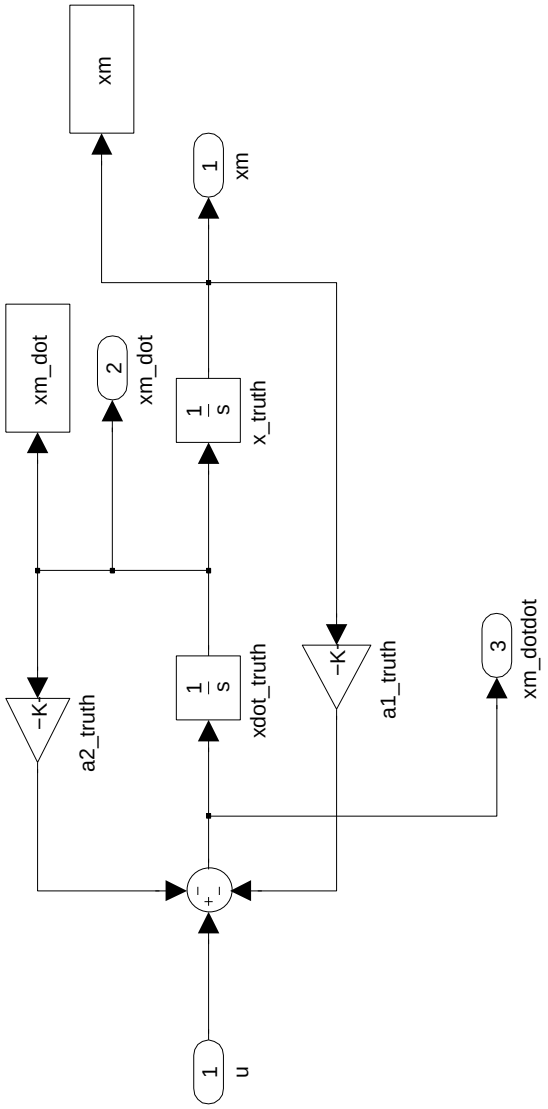

Figure A.8: Second Order System Model

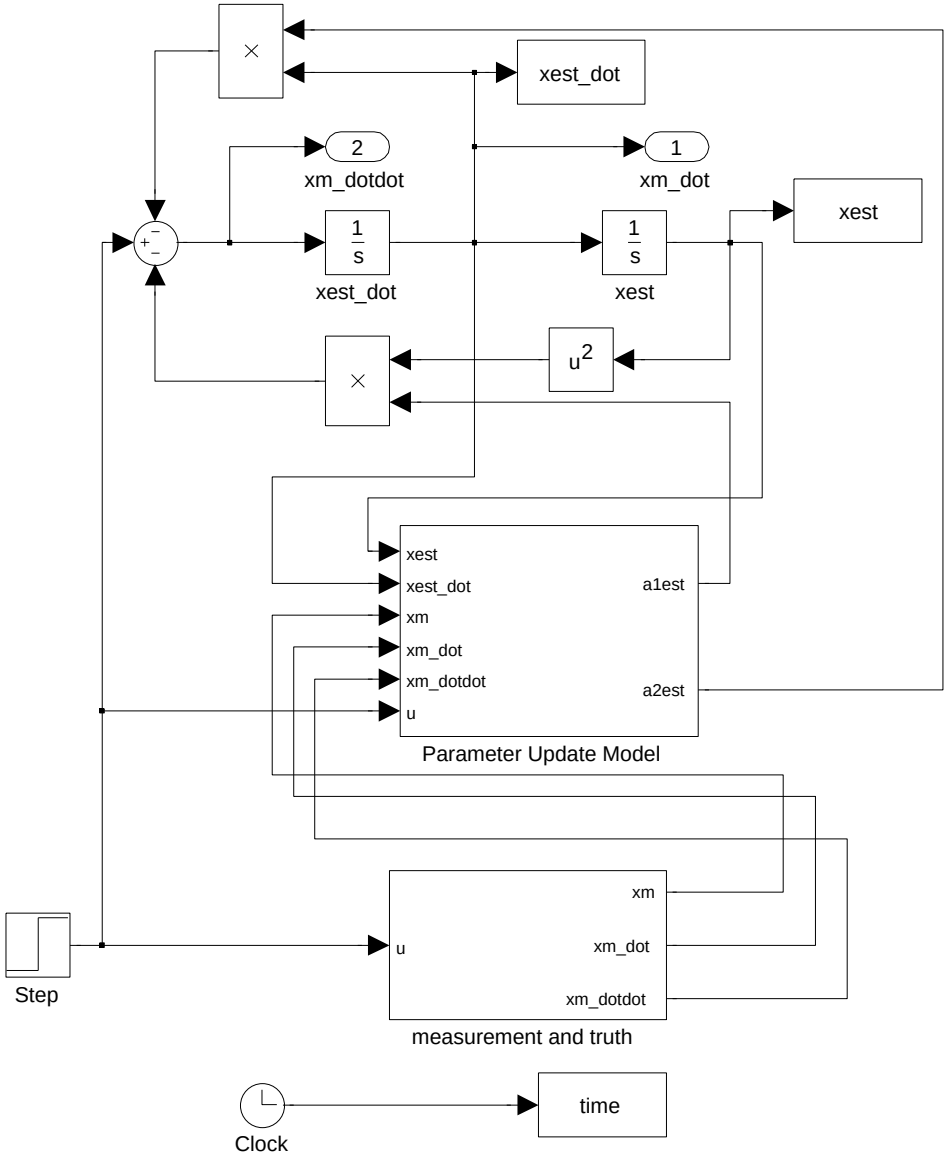

Figure A.9: Second Order Nonlinear Model Simulation Overview

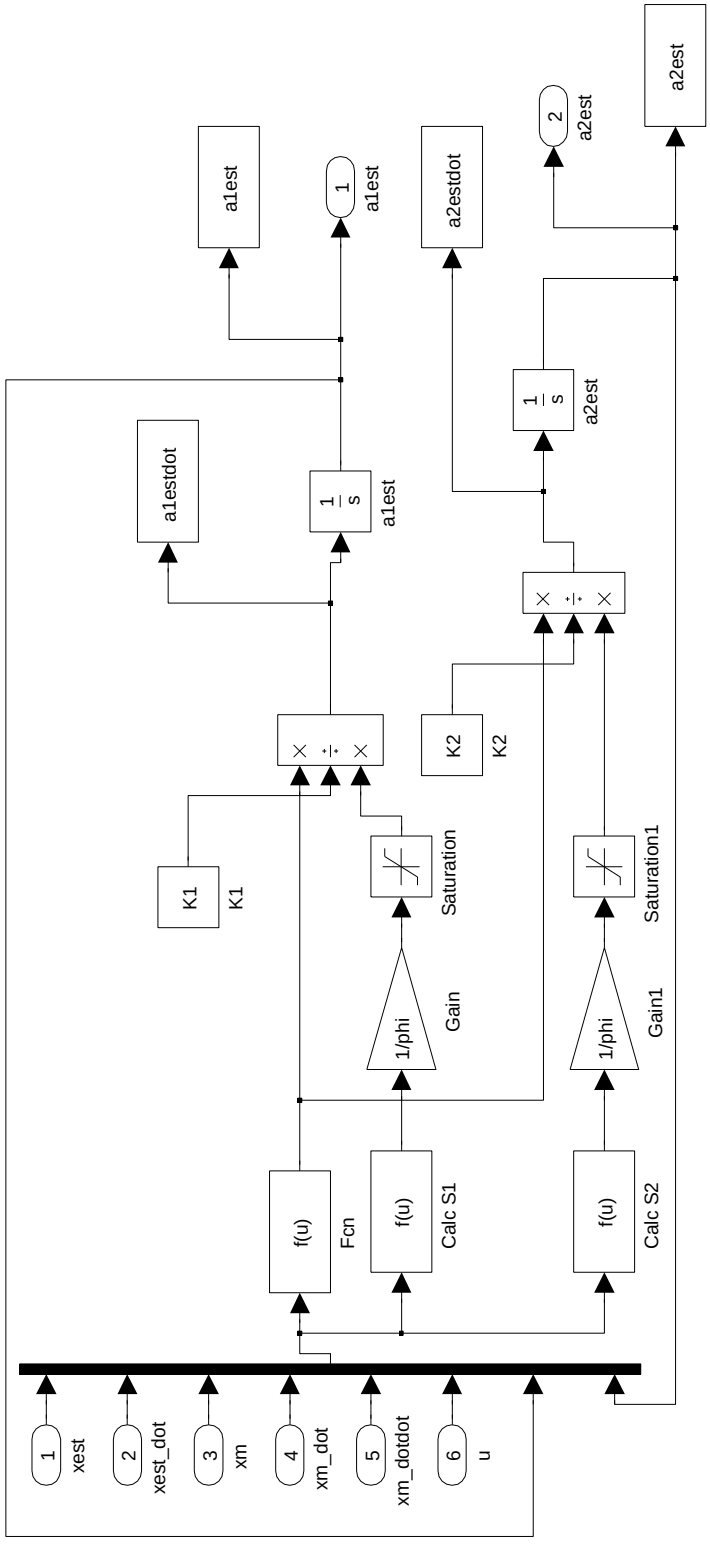

Figure A.10: Second Order Nonlinear Model Parameter Estimation Block

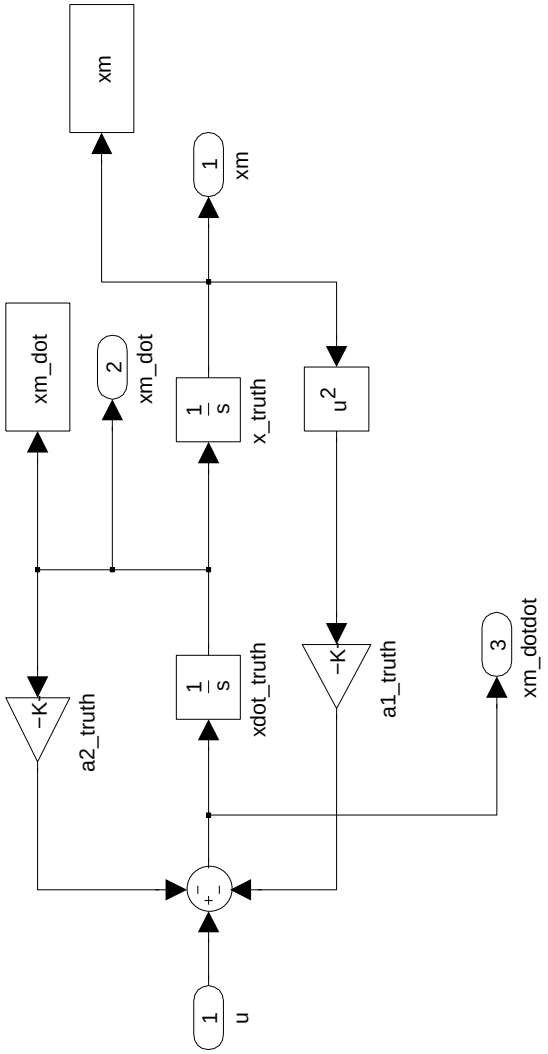

Figure A.11: Second Order Nonlinear System Model

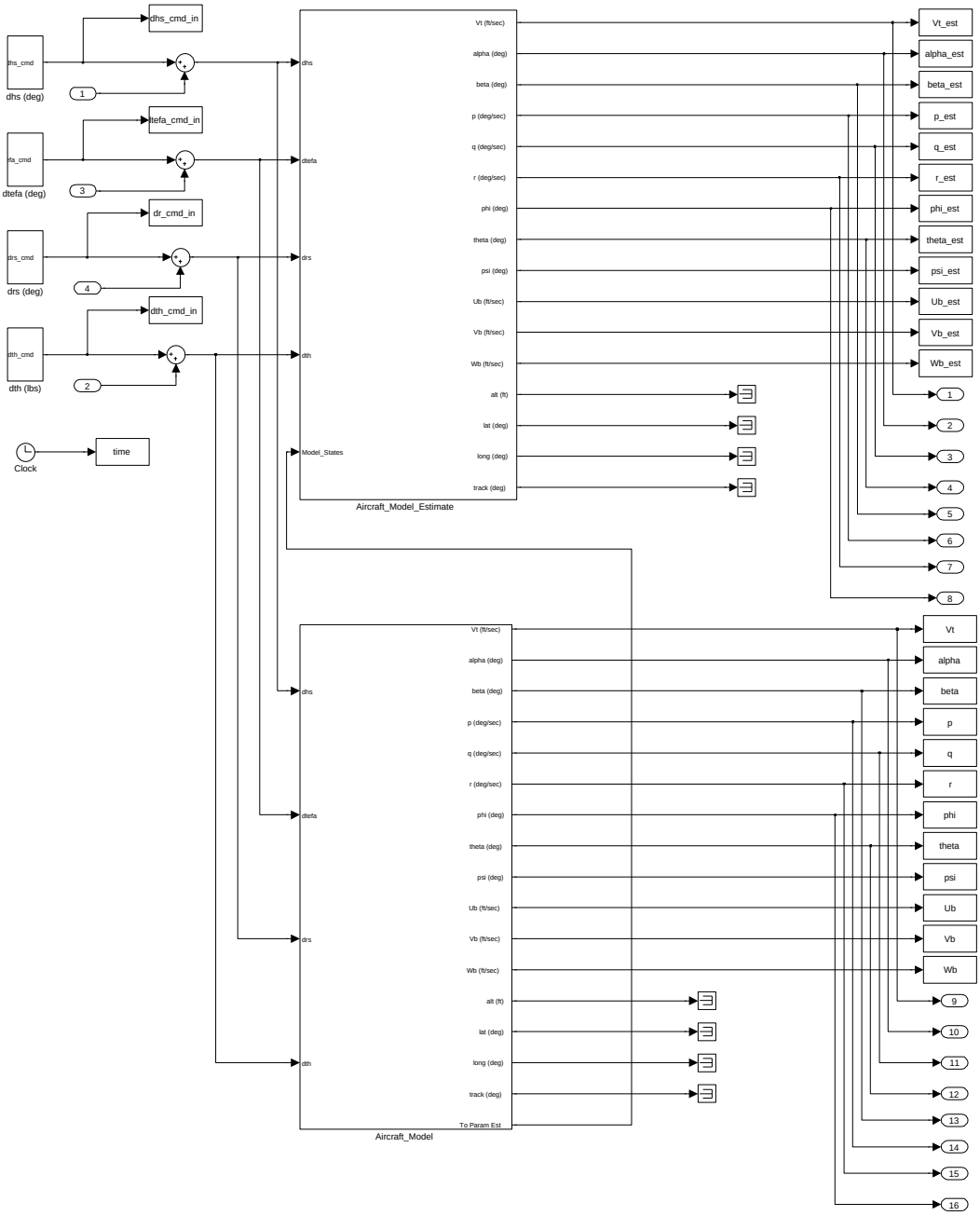

Figure A.12: Aircraft PID Simulation Overview

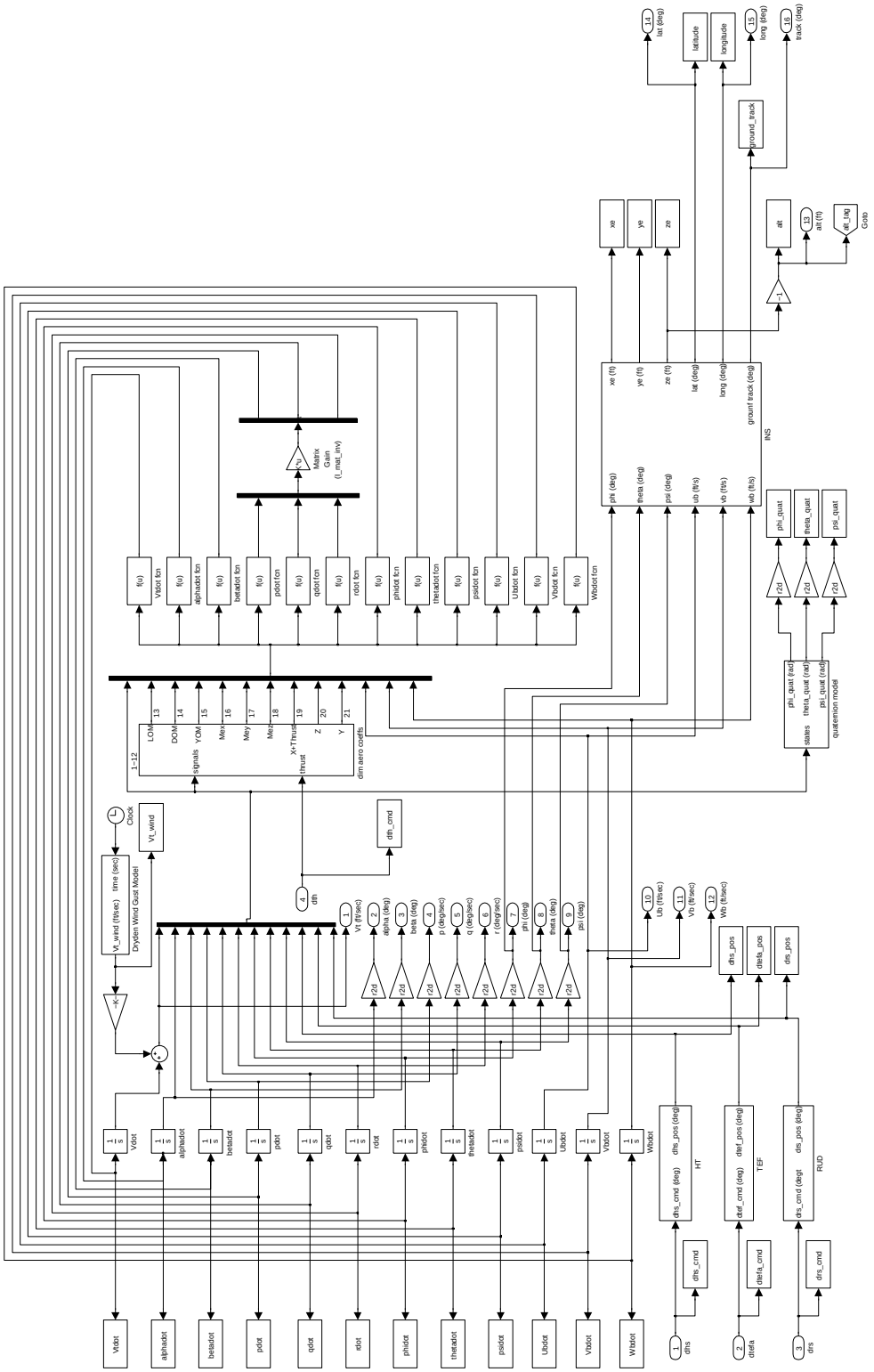

Figure A.13: Aircraft Model Overview

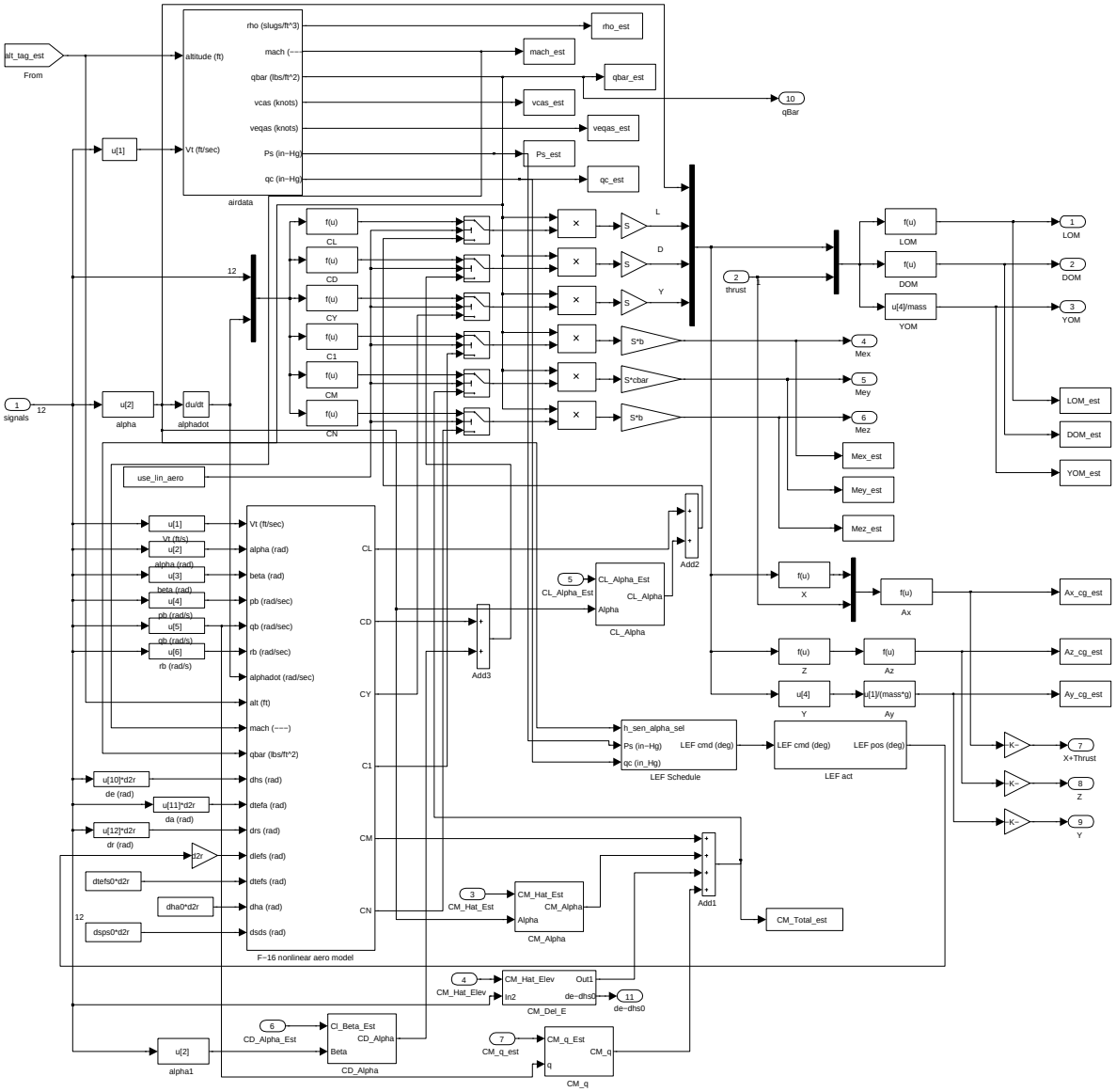

Figure A.14: F-16 Aircraft Aerodynamic Coefficient Calculation

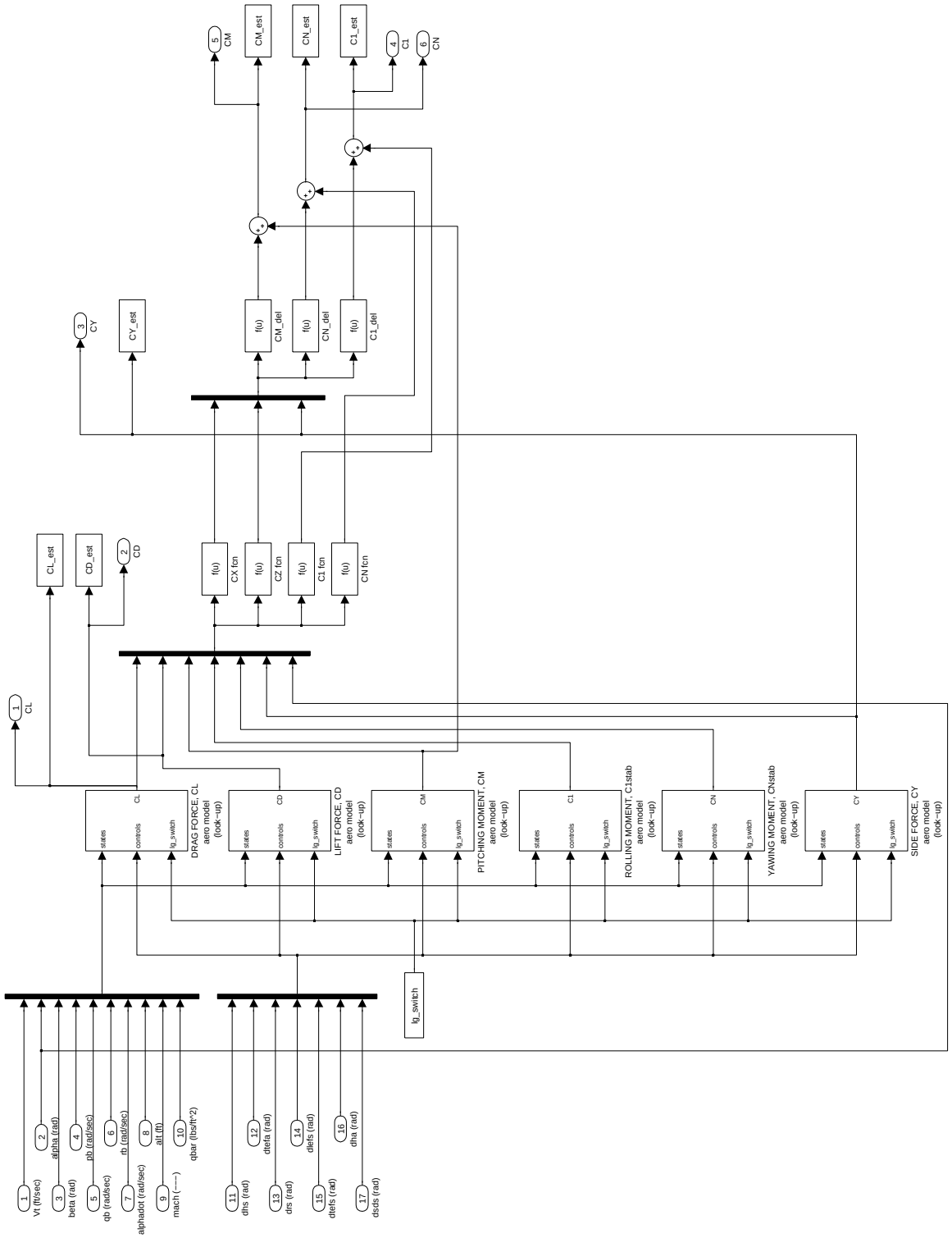

Figure A.15: F-16 Aircraft Nonlinear Aerodynamic Model

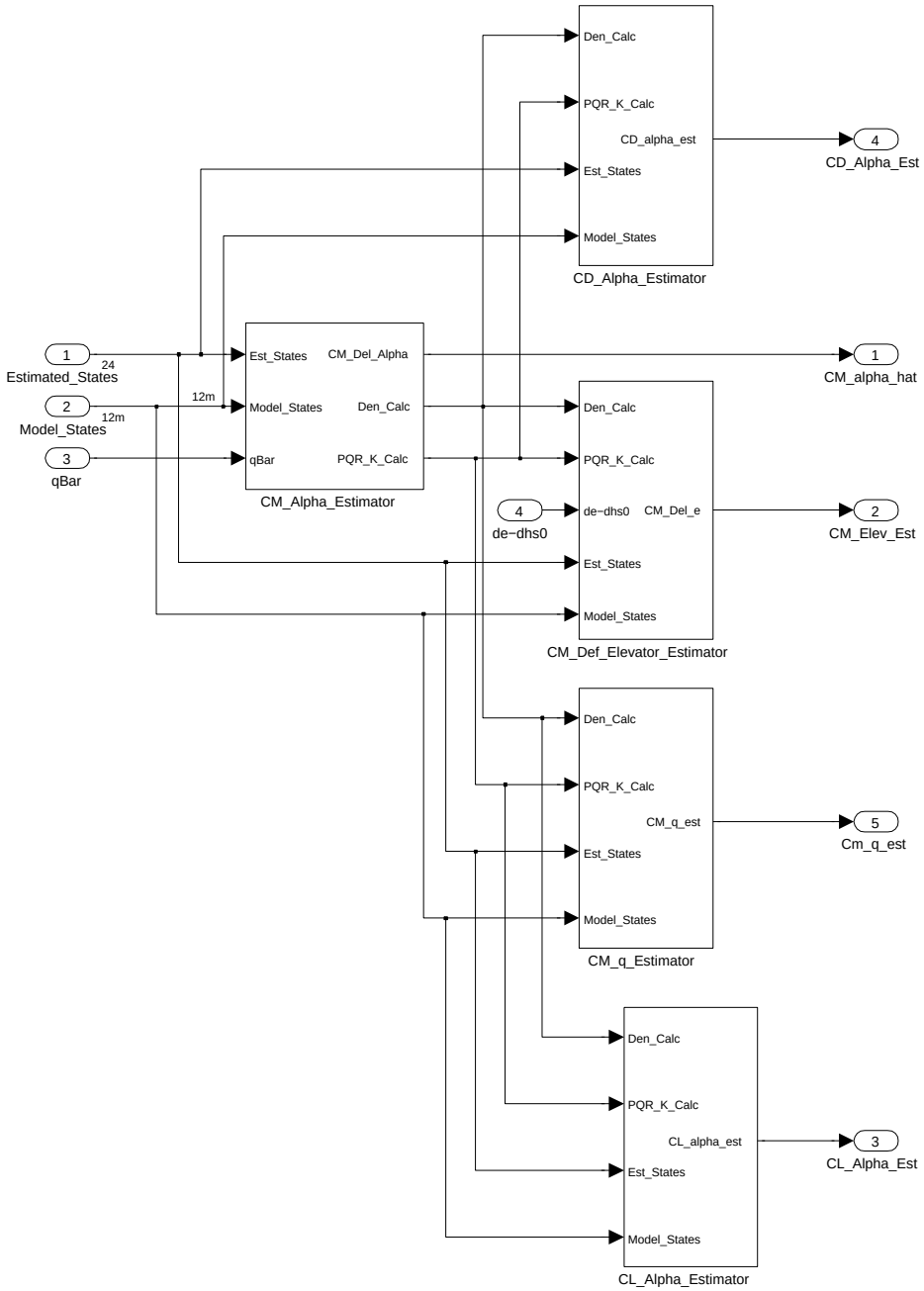

Figure A.16: Aircraft Parameter Estimation Block

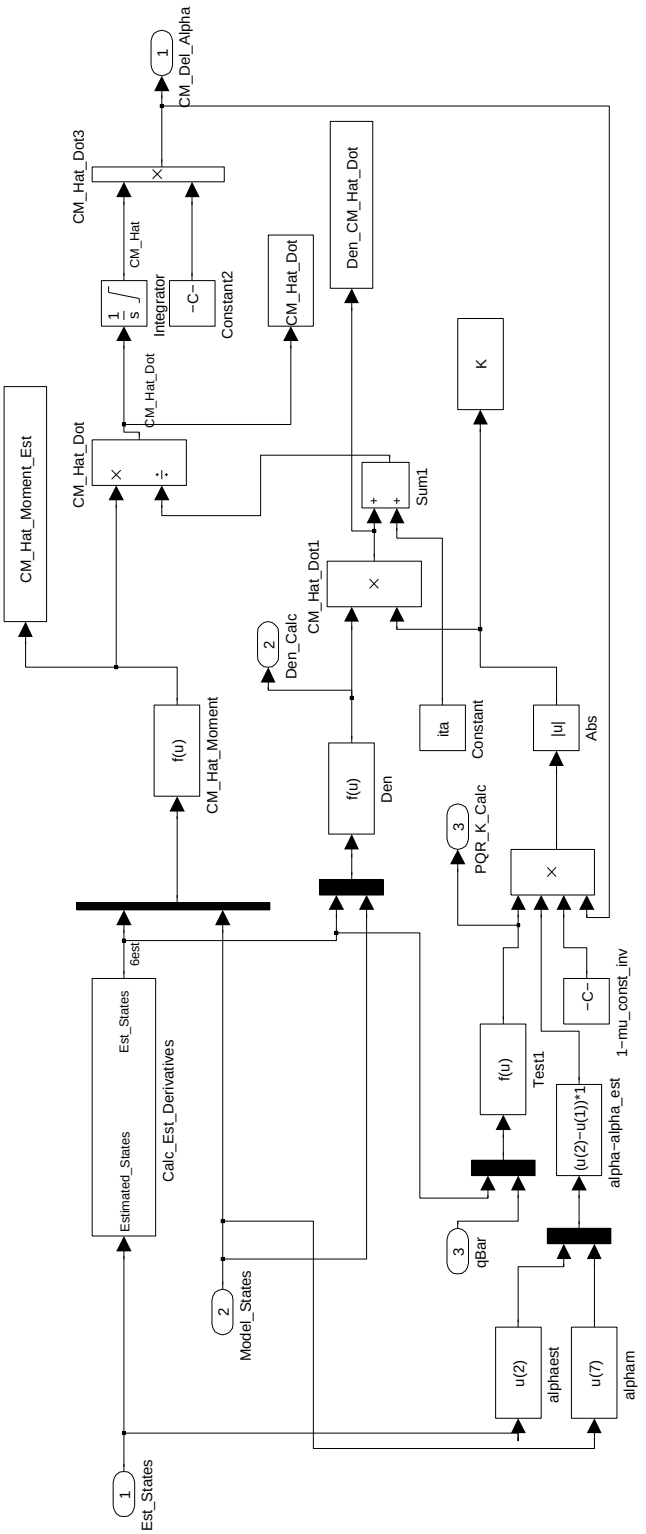

Figure A.17: C_{m_α} Estimation Overview

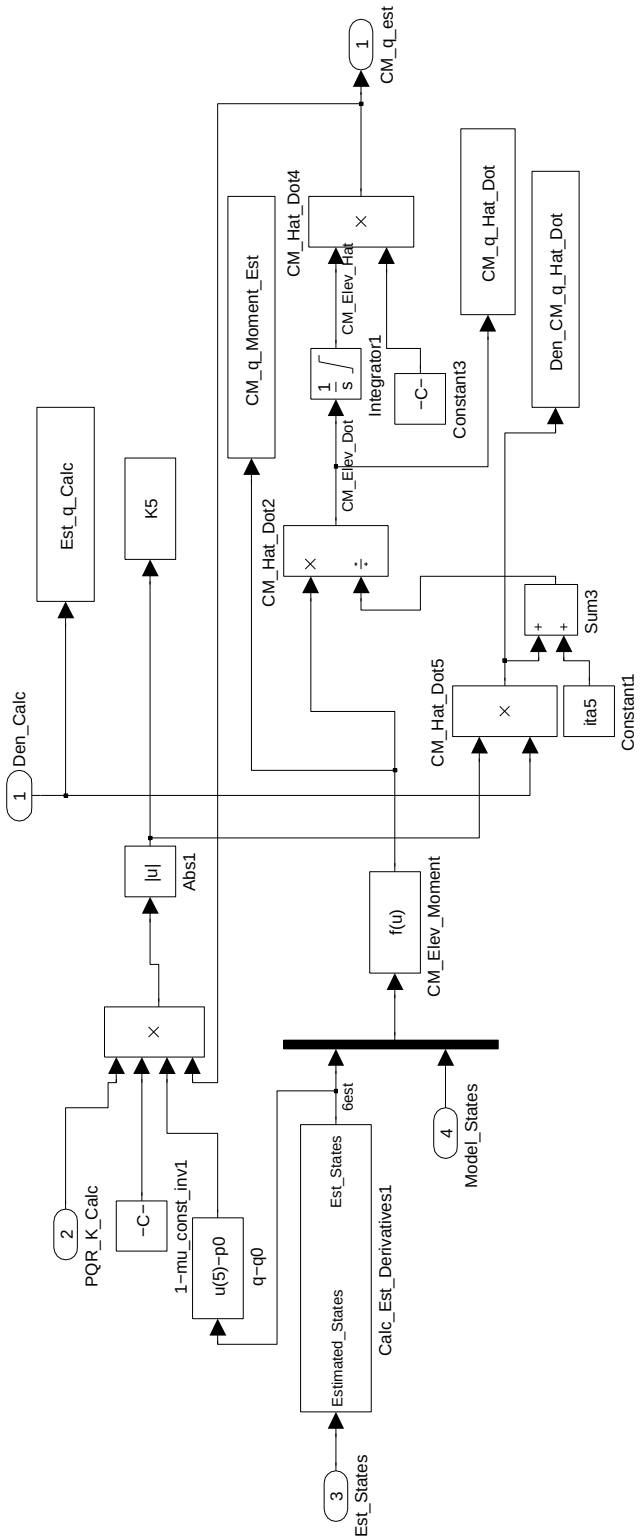

Figure A.18: C_{m_q} Estimation Overview